

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Infotehnoloogia eriala

Elen Kivi

Dokumentatsioon IT süsteemis

Magistritöö (30 EAP)

Juhendajad: Kaie Hansson

Vambola Leping

Autor: “.....” mai 2011

Juhendajad: “.....” mai 2011

..... “.....” mai 2011

Lubada kaitsmisele

Professor “.....” mai 2011

Sisukord

Sissejuhatus	3
1. Dokumentatsioon	5
1.1 Erinevad dokumendid	5
1.2 Dokumentatsiooni koostamise aeg	7
1.3 Dokumentatsiooni sihtgrupp	9
1.4 Dokumentatsiooni haldamine	12
2 Kohustuslik dokumentatsioon	14
2.1 Erinevad süsteemid	14
2.2 Standardid	16
3 Kaasajastatud dokumentatsioon	18
3.1 Dokumendi uuendamine	18
3.2 Versioneerimine	19
4 Uurimuse kirjeldus	22
4.1 Küsitluse ettevalmistamine ja läbiviimine	22
4.2 Arendaja küsimustiku tulemused	23
4.3 Kliendi küsimustiku tulemused	29
5 Tulemuste analüüs	36
6 Uurimuse võrdlus	44
Kokkuvõte	46
Summary	48
Viited	50
Lisad	54
Lisa 1, küsimustik arendajale	54
Lisa 2, küsimustik kliendile	55
Lisa 3, küsimustiku tulemused	58

Sissejuhatus

Tänapäeva maailmas on tarkvaraarenduse protsessis dokumentatsiooni kirjutamine aktuaalne teema, kuna see annab võimaluse fikseerida tehtud otsused ja seega lisab kindlust mõlemale osapoolle seoses valmiva IT süsteemiga. Lisaks võimaldab dokumentatsioon anda ülevaate, kuidas süsteem toimib ning kuidas seda kasutada. Sellest lähtuvalt toodavad kõik suured IT süsteeme arendavad ettevõtted dokumentatsiooni, kes vähem kes rohkem. Dokumentatsiooni võidakse kirjutada kõigis arenduse etappides, alates eelanalüüsist kuni juurutamiseni välja. Kuna paljudes projektides suureneb dokumentatsiooni hulk pidevalt, siis see, millise dokumendi koostamine on kohustuslik või tarvilik, muutub oluliseks. Sarnase tähtsuse omandab ka dokumentatsiooni kaasajastamine suure dokumentide hulga juures, kuna reaalses elus ei jõuta tihti kogu dokumentatsiooni regulaarselt uuendada.

Dokumentatsiooni kirjutamine tarkvaraarenduse käigus on pidev protsess, mis kuulub suure enamuse IT süsteemide arenduse hulka. Kuid peale pikemat uurimist selgub, et ühtsed nõuded, kuidas, kuna ja milleks dokumentatsiooni kirjutatakse puuduvad. Erinevates süsteemides ja ettevõtetes on antud teema lahendatud äärmiselt erinevalt. Magistritöö annab ülevaate IT süsteemide arenduses koostatavast dokumentatsioonist, kellele ja kuna see koostatakse. Lisaks kirjeldatakse, milline osa dokumentatsioonist on kohustuslik ja mis seda mõjutab ning uuritakse, milline on reaalne olukord Eestis. Töö teises pooles kirjutatakse, milliseid dokumente kaasajastatakse ja millised koostatakse süsteemi alustamisel ning kunagi ei uuendata.

Töö eesmärk on anda ülevaade dokumentatsiooni olemusest, koostamisest ja standarditest. Uurida, milline dokumentatsiooni hulk rahuldab tarkvara arendajat ja tellijat, millest tunneb enam puudu tellija ja millist dokumentatsiooni koostab või ei koosta arendaja. Uurimustöö käigus otsitakse ka vastuseid dokumentatsiooni kaasajastamist puudutatavatele küsimustele. Uurimuses tehtud järeldused võiksid olla sisendiks tellijale dokumentatsiooni tellimisel ja arendajale selle pakkumisel. Uurimus viiakse läbi Eesti tarkvara arendatavate ja tellivate ettevõtete seas. Enne uurimuse tegemist tehti laialdasi otsinguid antud teemal ning selgus, et mõned uuringud on koostatud, kuid Eesti kohta pole antud teemat uuritud.

Materjali leidmiseks kasutati märksõnasid nii eesti (dokumentatsioon, tarkvaraarendus, IT süsteem jne) kui inglise keeles (documentation, software engineering jne).

Lisaks püstitati hüpoteesid:

1. Tellija praktiline vajadus dokumentatsiooni järele erineb sellest, millist dokumentatsiooni arendaja teeb
2. Arendaja poolt koostatud oluline dokumentatsioon on aegunud ja seega ei vasta süsteemile
3. Süsteemi eripärad, nagu kestvus ja keerukus, mõjutavad dokumentatsiooni hulka

Järgnevalt tuuakse välja tähtsamad mõisted:

IT süsteem – süsteem, mis koosneb tarkvara süsteemist ning lisaks võivad kuuluda süsteemi ka erinevad seadmed

Rakendus – süsteem, mida kasutades kasutaja saab lahendada teabe töötlemisega seotud ülesandeid

Dokumentatsioon – koostatud dokument, mis kirjeldab, kuidas süsteem toimib, kuidas seda kasutada erinevates rollides. Dokumentatsioon ei ole ainult tekstiline kirjeldus, selle alla kuuluvad ka joonised, diagrammid jne.

1. Dokumentatsioon

Dokumentatsioon on dokument, mis kirjeldab süsteemi. See võib selgitada, kuidas süsteem toimib või kuidas seda kasutada erinevates rollides. Dokumentatsioon on suhtlusvahend erinevate osapoolte vahel. Lisaks on dokumentatsiooni funktsioon ka õpetada administraatoreid, kasutajaid ja teisi inimesi, kes on huvitatud tarkvara kasutamisest [1]. Dokumentatsiooni standardid IEEE (sh 829, 830) ja ISO (6592) defineerivad dokumentatsiooni kindlates olukordades. Näiteks IEEE 829 kirjeldab tarkvaraarenduse testimise dokumenteerimist, kuid IEEE 830 hoopis nõuete spetsifitseerimist.

Bill Curtis, Herb Krasner ja Neil Iscoe usuvad, et dokumentatsioon peaks kajastama kuidas nõuetega ja disainiga seotud otsused tehti, kuidas need teostati ning kuidas need süsteemi eluea jooksul muutuvad [2].

1.1 Erinevad dokumendid

IT süsteemi dokumentatsioon võib koosneda väga erinevatest dokumentidest. Juba 1980ndatel räägiti tarkvara dokumentatsiooni rohkusest [3]. Dokumente koostatakse IT süsteemi planeerides kui ka analüüsides. Dokumentatsiooni hulk suureneb tarkvara arendades veelgi, kuna arenduse, testimise ja juurutamise käigus koostatakse veel dokumente.

Projekti planeerimise faasis koostatakse projekti plaan, mida teeb tavaliselt projektijuht. Antud dokument sisaldab järgmisi osasid [4]:

1. Taustinformatsioon
2. Eesmärk
3. Sihtrühm
4. Tegevuskava
5. Teostamiseks vajalikud ressursid
6. Rahastamine

7. Juhtimise ja elluviimise organisatsioon
8. Oodatavad tulemused
9. Edukuse kirjeldus
10. Riskid

Lisaks eelnevale sisalduvad enamasti projekti plaanis või siis kirjeldatakse eraldi seisvate dokumentidena inimressursi plaan ja kommunikatsiooni plaan [5].

IT süsteemi nõudeid täpsustades kirjutatakse dokumentatsiooni, seal hulgas kirjeldatakse detailselt nõudeid. Esimesena pannakse paika äripoolle nõuded, selle järel kirjeldatakse süsteemi nõuded. Lisaks eelnevalt nimetatule kirjutatakse kasutuslugusid, milles täpsustatakse funktsionaalseid nõudeid [6], ja tehnilisi spetsifikatsioone. Peale nende võib kuuluda dokumentatsiooni hulka erinevaid jooniseid ja diagramme, näiteks: klassidiagramm, kasutuslugude skeemid (*ingl. k use case diagram*) jne. Süsteemi arhitektuuri paika pannes koostatakse arhitekti poolt arhitektuuri dokument, mis kirjeldab detailselt tarkvara arhitektuuri. Süsteemi disainer koostab disaini dokumendi, mis võib koosneda järgmistest osadest [7]:

1. Andmebaasi disain
2. Olemite kirjeldused
3. Kasutajaliidese disain
4. Turvalisuse disain
5. Vastastikmõjude diagramm
6. Seadistuste kirjeldused

Testija kirjutab tavaliselt testimise rapordi, testjuhtumid, testplaani [8]. Standard IEE 829, mis kirjeldab testimise jooksul kirjutatavat dokumentatsiooni lisab antud nimekirja ka disaini spetsifikatsiooni, protseduuri spetsifikatsiooni, testimise logid. Lisaks kirjutatakse kasutusjuhend, paigaldamise juhend ja taasteplaan.

Peale eelnevalt nimetatud dokumentidele, mis kirjeldavad otseselt tarkvara, mida luuakse või on tihedalt sellega seotud, kirjutatakse ka dokumentatsiooni plaani [1]. Antud dokument võib olla projektiplaani osa või eraldi seisev dokument. ISO 12207 standardis on samuti mainitud dokumentide nimekirja.

Kokkuvõtteks, erinevaid dokumente, mida näeb ette standard või mida mõni ettevõtte koostab on äärmiselt palju. Reaalses elus võib nii mõnigi dokument jääda koostamata,

kuna ei nähta selleks vajadust, lisaks võidakse koostada ettevõtte siseste tavade või nõudmiste järgi dokumente, mida eelnevalt ei nimetatud. Sellest kui palju aga Eestis tegutsevates tarkvara ettevõtetes reaalselt dokumentatsiooni koostatakse annab ülevaate uurimuse peatükk.

1.2 Dokumentatsiooni koostamise aeg

IT süsteemi arendus koosneb väga paljudest etappidest, mis võivad olla erinevad sõltuvalt valitud tarkvaraarenduse elutsükli mudelist. Tutvume esmalt veidi lähemalt tarkvaraarenduse elutsükliga ning seejärel vaatame, millist dokumentatsiooni kirjutatakse tarkvaraarenduse erinevates etappides.

Tarkvaraarenduse elutsükkel algab süsteemi planeerimisega ning lõpeb süsteemi juurutamise ning hooldamisega. Elutsükli mudel annab juhised, kuidas organiseerida, planeerida personali, eelarvet, ajagraafikut ja juhtida projekti erinevas keskkonnas.

ISO 12207 standard käsitleb tarkvara elutsükli protsessi pakkudes raamistikku alates idee tekkimisest, mille lahenduseks on kas osaliselt või tervenisti tarkvara, ja ulatudes rakenduse töö lõppemiseni. Standard jagab tarkvara elutsükli kolme suuremasse protsessi:

1. Primaarprotsessid
2. Abiprotsessid
3. Organisatsioonilised protsessid

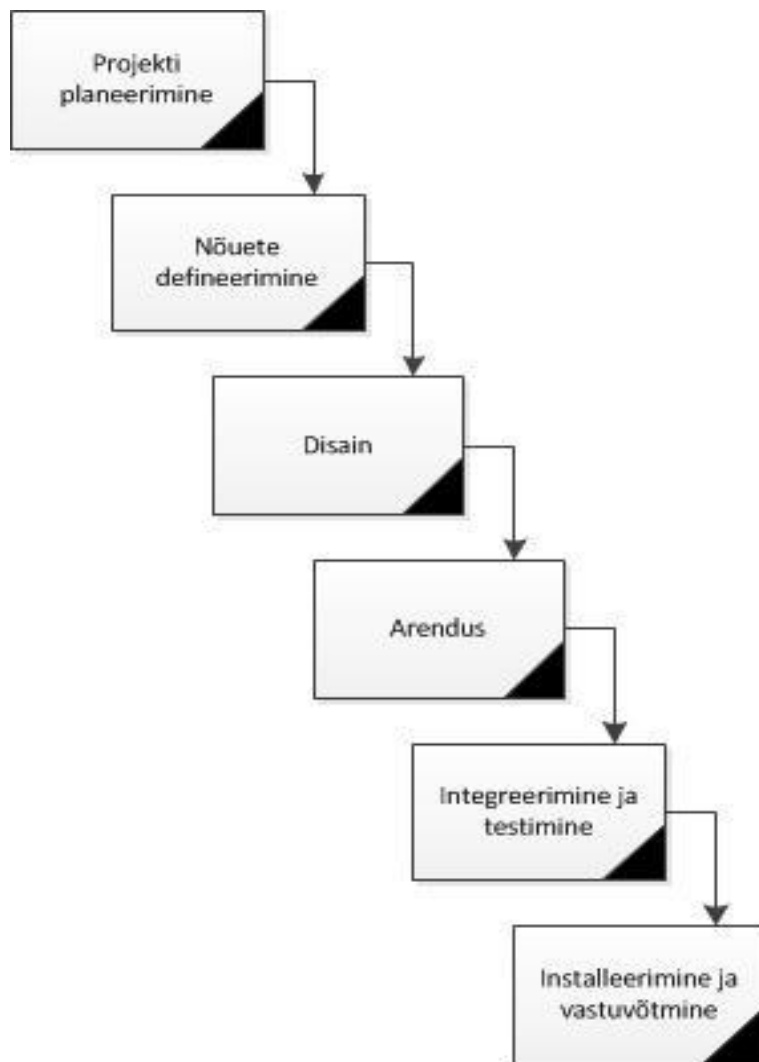
Kõik need protsessid jagatakse veel omakorda väiksemateks osadeks. Dokumenteerimine kuulub abiprotsesside hulka, kuid see toimub arenduse käigus, mis on primaarprotsess.

IT maastikul on kasutusel mitmeid erinevaid tarkvaraarenduse elutsükli mudeleid. Klassikaline tarkvaraarenduse elutsükli mudel sisaldab tavaliselt mõnda järgmistest tegevustest [9]:

1. Süsteemi planeerimine/algatamine
2. Nõuete analüüs ja spetsifitseerimine
3. Funktsionaalsete nõuete kohta prototüübi koostamine
4. Arhitektuuri disaini ja konfiguratsiooni spetsifitseerimine

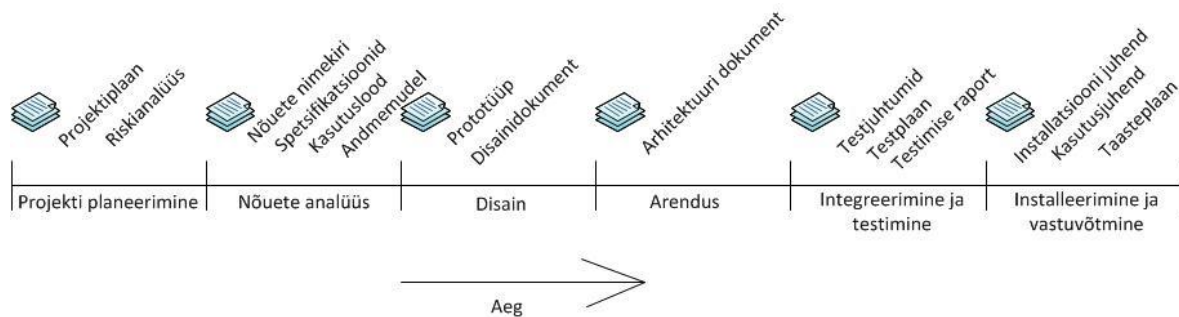
5. Detailne komponentide disaini spetsifitseerimine
6. Komponentide arendamine ja vigade parandamine
7. Tarkvara integreerimine testimine
8. Dokumentatsiooni läbivaatus ja süsteemi üleandmine
9. Paigaldus ja installeerimine
10. Koolitus ja kasutamine
11. Süsteemi hooldus

Samas üks tuntumaid elutsükli mudeleid, kaskaadmudel [*ingl k. Waterfall mudel*], jaotab tarkvaraarenduse elutsükli ainult 6 erinevasse etappi, joonis 1. Kuna kaskaadmudeli näol on tegemist ühe palju kasutatud tarkvaraarenduse elutsükli mudeliga, siis autor vaatabki dokumentatsiooni koostamist tuginedes antud mudelile.



Joonis 1, Kaskaadmudel [10]

Esimeses etapis, projekti planeerimine, koostab projektijuht projektiplaani ja riskianalüüsi. Nõuete analüüsi etapis koostab süsteemianalüütik loodava süsteemi nõuete nimekirja, spetsifikatsioonid, kasutuslood. Lisaks eelnevalt nimetatule koostatakse analüüsi käigus enamasti ka andmemudel. Kui eelnevalt nimetatud dokumendid on valminud, siis asutakse järgmisena koostama prototüüpi ja disainidokumenti disaini faasis. Arenduse käigus kirjutatakse arhitektuuri dokument ning selle järel koostab süsteemi testija testjuhtumid, testplaani ja raporti testimise kohta.



Joonis 2, ajatelg

Viimases tarkvaraarenduse etapis kirjutatakse installatsiooni juhend ja kasutusjuhend. Tavaliselt koostatakse selles faasis ka lõppkasutajale mõeldud kasutusjuhend, kuid see võidakse kirjutada ka enne, näiteks analüüsi või testimise etapis. Osades projektides koostatakse ka taasteplaan, kas siis arenduse viimases faasis või varem.

Dokumentatsiooni kirjutamine on tihedalt seotud arenduse etapiga, kuigi sellest hoolimata võib reaalses tarkvaraarenduses olla erandeid. Kokkuvõtteks koostatakse igas arenduse etapis sellele omast dokumentatsiooni, mis on oluline just selle faasi seisukohalt.

1.3 Dokumentatsiooni sihtgrupp

Dokumenti kirjutades tuleb mõelda, kes seda hiljem lugema hakkab, vastavalt sellele tuleb kohandada kirjastiil ja sisu. Tarkvaraarenduses koostatavat dokumentatsiooni kirjutatakse erinevatele sihtgruppidele ja samuti erinevad ka eesmärgid [11]:

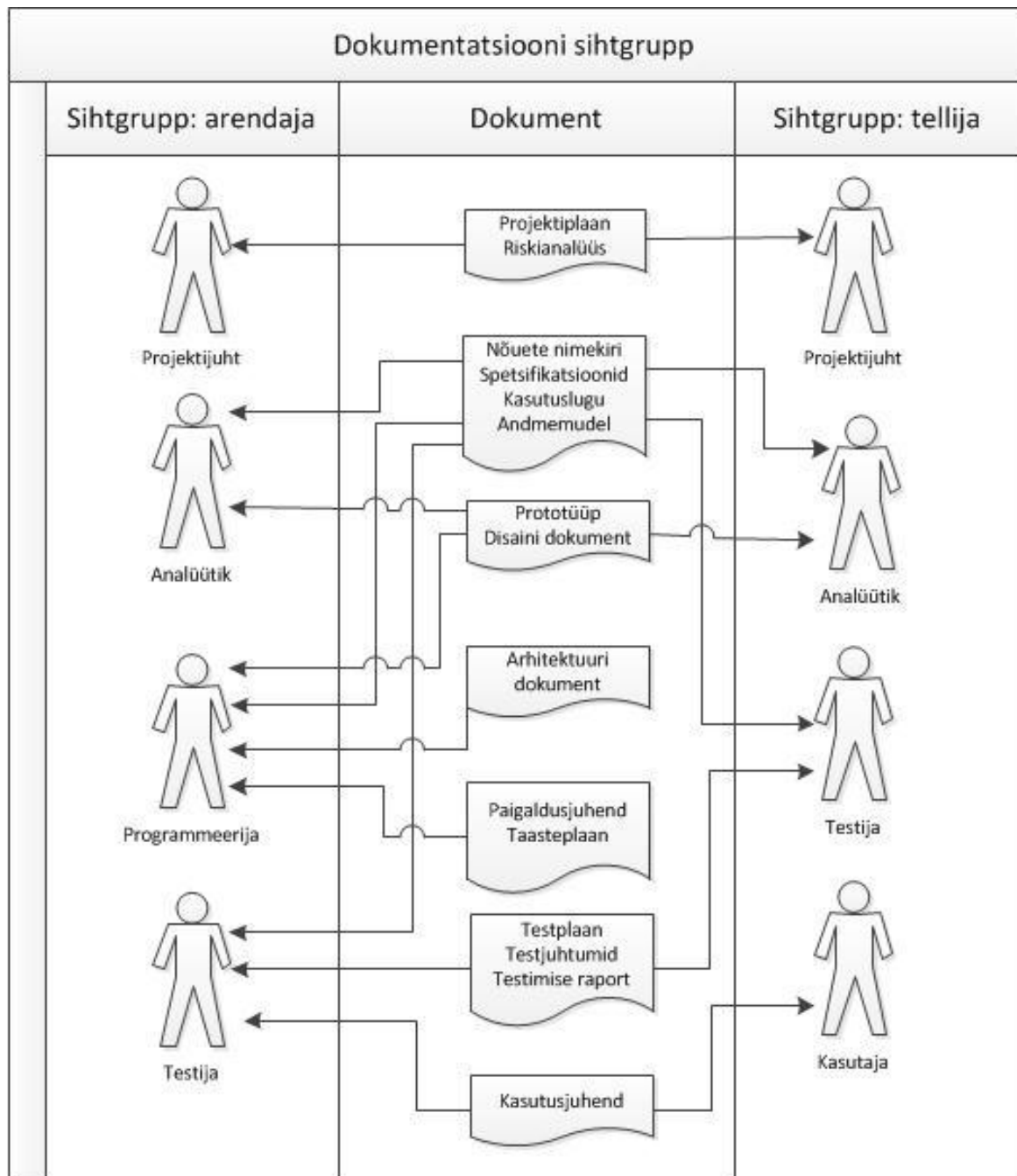
1. Suhtlusvahend meeskonna liikmete vahel k.a. kliendi

2. Süsteemi informatsiooni repositoorium (nt arhitektidele vajalik)
3. Informatsiooni allikas juhtidele planeerimisel
4. Abiks kasutajatel süsteemi kasutamisel ja administreerimisel

Osa dokumentatsioonist on mõeldud tellijale ehk kliendile lugemiseks, näiteks kasutuslood. Lisaks võib olla kasutusloo sihtgrupiks ka arendaja, vt joonis 3. Mõned tarkvaraarendajad koostavad siinkohal kaks eraldi kasutuslugu, üks kliendile, teine arendajale, kuna info, mida need osapooled antud dokumendist otsivad, on erinev. Samas oma kogemuste põhjal saan väita, et osad ettevõtted aga kirjutavad jälle vastupidiselt ainult ühe versiooni kasutusloost, mis on mõeldud lugemiseks nii arendajale kui kliendile. Sellisel juhul võidakse eraldada dokumendis sisuline ja tehniline osa, pannes üks neist näiteks sulgudesse või kasutades erinevaid kirjastiile. Lisaks võivad sama dokumenti kasutada ka teised osapooled peale kliendi ja programmeerija, näiteks testija. Spetsiaalselt küll enamasti pole kasutuslugu mõeldud tarkvara testijale, kuid selle põhjal saab testija selgeks, kuidas mingi komponent peab toimima.

Teiselt poolt vaadates leidub aga dokumentatsiooni, mis on mõeldud ainult ühele grupele, näiteks lõppkasutajale, kuid mida peab lugema ka testija või teised arendaja meeskonna liikmed. Kasutusjuhend on üks selline dokument, mis kirjutatakse spetsiaalselt süsteemi kasutajale, kuid selle lugemine on otstarbekas ka testijale, kes peab süsteemi kasutama testimiseks või arendajale, kellel tuleb testida enda kirjutatud komponenti ja mõnikord selleni jõudmiseks teha süsteemis eelnevalt mitmeid samme.

Kolmandaks on dokumente, mida kirjutab arendaja enda meeskonnale. Arhitekt kirjeldab detailselt loodava süsteemi arhitektuuri, et edaspidi oleks võimalik arendajatel sellega arvestada koodi kirjutades. Enamasti arhitektuuri dokumenti klient ei loe, samuti tutvub sellega harva testija. Sarnast eesmärki täidab näiteks testija poolt kirjutatud testjuhtumite dokument, mis on testimisprotsessi artefakt ja sisaldab detailset kirjeldust, kuidas kontrollida testitava objekti vastavust mingile nõudele [12]. Enamasti on antud dokument mõeldud testijatele endile, kaasa arvatud tellija poolsetele. Samas loeb harva seda dokumenti analüütik või lõppkasutaja.



Joonis 3, dokumendi sihtgrupp

Enamasti kirjutab tarkvaraarenduse igas etapis dokumentatsiooni erinev inimene. Analüüsi faasi dokumentatsiooni paneb kokku analüütik, kuid arhitektuuri dokumendi arhitekt, sarnaselt on see tarkvaraarenduses olnud juba 1990ndatest, kus samuti erinevad spetsialistid vastutasid erinevate dokumentide eest [13].

IT süsteemi dokumentatsiooni sihtrühmaks on tarkvaraarenduse meeskonna liikmed, klient ja kasutaja. Muidugi võivad arenduse käigus loodud dokumentatsiooni lugeda ka erinevatel põhjustel teised sihtgrupid, näiteks audiitor, kuid dokumenti kirjutades on mõeldud just eelnevalt nimetatud gruppidele.

1.4 Dokumentatsiooni haldamine

Dokumentatsiooni haldamine muutub aina rohkem päevakajaliseks teemaks, kuna dokumentide hulk kasvab ja nende säilitamine ja haldamine on keerukas. Dokumentatsiooni haldamiseks luuakse süsteeme, mis on kas eraldi seisvad või ettevõtte sisuhaldussüsteemi osad, et salvestada elektroonilisi dokumente. Samuti kasutatakse dokumentide hoidmiseks juba olemasolevaid dokumendihaldussüsteeme.

Hea dokumendihaldustarkvara annab organisatsioonile palju eeliseid. Näiteks võimaldab see süsteem lihtsasti otsida elektroonilisi dokumente erinevate parameetrite järgi. Lisaks annab dokumentatsiooni haldamise tarkvara kindlust, kuna kogemata kustutatud versioonid on võimalik taastada. Seoses versioneerimisega on dokumendihaldus süsteemist lihtne jälgida, milline versioon on kõige uuem ja kes on millise versiooni autor. Oluline lisafunktsionaalsus on dokumentide lukustamine, see tähendab, et korraga saab muuta sama dokumenti ainult üks kasutaja. Selline süsteemi käitumine välistab võimaluse, kus sama dokumenti muudavad korraga mitu kasutajat ja salvestamise käigus jääb alles, vaid ühe kasutaja poolt tehtud muudatused. Veel võimaldab enamus dokumentatsiooni haldamise süsteeme jagada kasutajatele õigusi kaustade täpsusega ehk on võimalik piirata ligipääs osadele dokumentidele [14].

Olemasolevaid dokumendihaldustarkvarasid on kasutusel väga erinevaid, näiteks: Alfresco, Postipoiss, Amphora, LiveLink jne. Amphora on juhtivaid digitaalse asjaajamise ja dokumendihalduse tarkvarasid Eestis [15]. Amphora on võimalik teostada järgmisi olulisemaid tegevusi:

1. Laadida dokumente üles
2. Luua dokumente

3. Jälgida dokumentide ringlust jne

Veebipõhine dokumendihaldustarkvara Postipoiss on kasutusel Eestis alates 1998. aastast [16] ning pakub väga palju võimalusi:

1. Dokumentide registreerimine
2. Otsingud
3. Dokumentidega seotud tööülesanded
4. Aruanded jne

Alfresco on dokumendihaldustarkvara avatud platvormil, mis on samuti leidnud laialdast kasutamist. Süsteem võimaldab hallata kõiki organisatsiooni digitaalseid dokumente kogu dokumendi elutsükli jooksul. Lisaks kuuluvad Alfresco funktsionaalsuse hulka ka otsing ja tööülesanded.

LiveLink on dokumentide ja teabehalduse lahendus suurtele ja keskmistele organisatsioonidele [17]. Selle dokumendihaldusetarkvara funktsionaalsuse hulka kuuluvad:

1. Dokumentide haldamine
2. Otsingud
3. Teavitused
4. Dokumentide elutsükli haldamine jne

Kokkuvõtteks on dokumentide haldamine lahutamatu osa tarkvara dokumentatsiooni loomisest ning selleks on loodud mitmeid erinevaid süsteeme.

2 Kohustuslik dokumentatsioon

Erinevates IT süsteemide projektides võib näha väga erinevat dokumentatsiooni, mida koostatakse. Mõnes süsteemis ei koostata testraportit, teises aga peetakse seda väga oluliseks. Süsteemi eripära, tellija ja arendaja, järgitavad standardid, lepingud, traditsioonid jne mõjutavad, milline on kohustuslik dokumentatsioon. Aspekte, mis võivad anda ühele dokumendile suure väärtuse ja vastupidiselt muuta teise dokumendi ebaoluliseks on mitmeid. Mis määratleb dokumentatsiooni, mis peab olema?

2.1 Erinevad süsteemid

Dokumentatsioon, mis on ühes süsteemis kohustuslik, kuid teises mitte, võib olla määratud süsteemi eripäradega. Erinevad IT tarkvara nüansid nagu kasutajate arv või süsteemi eluiga muudavad dokumentatsiooni väärtust palju.

Esimene aspekt, mis määratleb kohustusliku dokumentatsiooni, on leping. Tellija kirjutab välja kuulutatud hankes, millised dokumendid tuleb üle anda koos tarkvaraga. Riigi Infosüsteemide Arenduskeskuse lehel tutvudes riiklike hangetega on näha, kus mõnes hankes pole üldse kirjeldatud kohustuslikke dokumente, samas leidub neid hankeid, kus on detailselt kirjeldatud dokumendid, mis tellija peab esitama. Näiteks Põllumajanduse Registrite ja Informatsiooni Ameti andmeaida ja aruandlussüsteemi hankes on nõutud tellijalt täpset dokumentatsiooni. Hankes on kirjutatud, et analüüsi tulemiks on analüüsi dokumentatsioon, mis sisaldab järgmisi dokumente:

1. Loodava tarkvara funktsionaalsed- ja mittefunktsionaalsed nõuded
2. Loodava tarkvara valideerimise- ja aktsepteerimise nõuded
3. Loodava tarkvara kasutuslood
4. Loogiline ehk kontseptuaalne andmemudel

Lisaks on kirjeldatud ka teiste arenduse etappide tulemid nagu disaini protsessi, testimise, juurutamine jne.

Teiseks võib määrata kohustusliku dokumentatsiooni arendatava süsteemi eesmärk. Tarkvara, mille eesmärk on jagada informatsiooni, kus midagi muuta ega lisada kasutajad ei saa, pole kriitiline süsteem. Muidugi võib see tekitada kahju, kui andmed on näiteks vigased sellises süsteemis, kuid enamasti pole see elulise tähtsusega. Samas on IT süsteeme, mille eksimuse või töötamast lakkamise tagajärjel võib saada keegi viga või võib kahju ulatuda mitmetesse miljonitesse eurodesse, näiteks meditsiini või lennundusega seotud süsteemid. Väiksemat töökindlust nõudvates süsteemides ei ole dokumentatsioonil nii suurt kaalu, ilmselt pole kasutusjuhendi koostamine kohustuslik, samas kui vaadata väga kriitilisi süsteeme, siis neis on kasutusjuhendi koostamine möödapääsmatu, kuna kasutajate eksimine pole taunitav.

Sarnane olukord kriitiliste ja vähem kriitiliste süsteemidega võib olla ka ühe organisatsiooni siseselt. Osad süsteemid on tähtsamad ja teised on toetava funktsiooniga ning nende puhul ei nõuta nii palju dokumentatsiooni. Baassüsteemid, mis on organisatsioonile olulise väärtusega, on rohkem ja põhjalikumalt dokumenteeritud. Uurimuse käigus tuli välja, et see erinevus võib olla ühe ettevõtte sees äärmiselt suur. Näiteks üks infosüsteem on klientide teenindamiseks, mis on baassüsteem ja teine dokumentide haldamiseks, mis on vähem kriitiline ja seega vähem dokumenteeritud.

Dokumentatsiooni järele on vajadus väiksem väga lihtsate ja lühiajaliselt kestvates projektides vastupidiselt keerulise äriloogikaga ja pika elueaga süsteemides. Kui süsteemi arendamise aeg on pikk, siis on kohustusliku dokumentatsiooni hulk suurem, kuna inimesed ei mäleta enam tehtud otsuseid ja nende põhjuseid. Näiteks kui puudub kasutuslugu komponendi kohta, mida arendati projekti alguses ning hiljem ilmneb vajadus seda komponenti täiendada, siis ei tea enam keegi täpselt, kuidas see töötab ja miks see just nii peab töötama. Sama probleem on väga keerulise äriloogikaga, mis võib ununed ja seega peab olema täpselt dokumenteeritud. Lihtsates süsteemides, aga ei ole vajadus detailse kasutusloo järgi nii suur, kuna äriloogika on lihtne. Heaks näiteks on erinevad registrid, mille eesmärk on avalikustada andmeid ning kasutajal ei lubata muuta ega lisada midagi.

Lisaks mõjutavad kohustuslikku dokumentatsiooni süsteemi lõppkasutajad. Paljude kasutajatega projektides on dokumentatsiooni roll märksa suurem kui süsteemides, mis on mõeldud organisatsiooni siseseks kasutamiseks. Kui tarkvara kasutatakse ühe asutuse siseselt ning väikese arvu inimeste poolt, siis ei ole keeruline kasutajale lihtsalt selgitada,

kuidas süsteemi kasutada ja seega võib olla kasutusjuhend üldisem või lausa puududa. Vastupidine näide on aga avalikkusele mõeldud rakendustes. Keegi meist ei kujuta ette, et kodus mõnda rakendust kasutades seisaks arendaja meie kõrval ja selgitaks, kuidas süsteemi kasutada. Samuti on oluline kasutusjuhend väga suurtes ettevõtetes, kui töötajaid on sadu või enam. Lisaks kasutajate arvule mõjutab veel dokumentatsiooni see, kui kasutajaid on vähe ja nad asuvad erinevates kohtades, sellises olukorras on kasutusjuhendi olemasolu vajalik.

Kokkuvõtteks, kohustuslikku dokumentatsiooni ei ole erinevates süsteemides üheselt võimalik määrata. Erinevatest rakenduse, arendaja, tellija ja lõppkasutaja eripäradest on kirjutatav dokumentatsioon varieeruv.

2.2 Standardid

Mitmetes tarkvaraarendust puudutavates standardites kirjeldatakse dokumentatsiooni. Mõnedes täpsemalt kui teistes, kuid samas on enamik neist standarditest ühe kindla etapi kohta tarkvara arendamise protsessis, näiteks testimise kohta, kuid tervikliku dokumenteerimise protsessi kohta ei leidu detailseid standardeid. Kas leidub standardeid, kus kirjutatakse, milline on kohustuslik dokumentatsioon tarkvaraarenduses?

Autor vaatab siinkohal paari standardit. Esimesena vaadeldakse ISO 12207 - "Tarkvara elutsükli protsessid" [18] standard, kus kirjeldatakse dokumenteerimist kui väga olulist protsessi. Standardis märgitakse, et esimese dokumendina tuleb koostada nimekiri dokumentatsiooni osadest, mis on olulised loodava tarkvara seisukohalt. Lisaks märgitakse ära, et dokumentide arv ei tohiks kasvada liiga suureks ja vältida tuleb puuduliku dokumentatsiooni, kuid detailsemalt ei kirjeldata, millised dokumendid on soovituslik koostada, välja arvatud nimekiri dokumentatsiooni osadest.

Järgmisena vaadatakse detailsemat standardit, mis käsitleb ühte tarkvaraarenduse etappi, mitte terviklikku protsessi. Näiteks ANSI/IEEE 829 on standard tarkvara testimise etapis kirjutatava dokumentatsiooni kohta. Standardis tuuakse välja järgmised dokumendid [19]:

1. Testimise plaan (*ingl. k test plan*)
2. Testimise detailprojekt (*ingl. k test-design spesifikation*)
3. Testide spetsifikatsioon (*ingl. k test-case spesifikation*)
4. Testimise protseduuri spetsifikatsioon (*ingl. k test-procedure specification*)
5. Testitavate objektide üleandmise aruanne (*ingl. k test-item transmittal report*)
6. Testimise käik (*ingl. k test-log*)
7. Testprobleemide aruanne (*ingl. k test-incident report*)
8. Testimise kokkuvõte (*ingl. k test-summary report*)

Seega võib järeldada, et antud standardi järgi on kohustuslik koostada kõik ülevalpool nimetatud dokumendid.

Maailmas on loodud standardeid kõikide põhiliste elutsükli tegevuste dokumenteerimise kohta [20], seega on standartidega määratletud soovituslik dokumentatsioon igas arenduse etapis.

3 Kaasajastatud dokumentatsioon

Tarkvaraarenduses on dokumentatsiooni kaasajastamine väga aktuaalne teema, dokumentide hulk võib olla väga suur ning selle pidev uuendamine on arendaja meeskonnale väga palju aega nõudev ja seega tellijale äärmisel kulukas protsess. Dokumentatsiooni uuendamist nõuab aga uute nõuete lisandumine, mille tõttu tuleb muuta olemasolevat funktsionaalsust või hoopis lisada juurde. Samas on aegunud dokumentatsioon kasutu, kuna edastab vale informatsiooni süsteemi kohta ja võib põhjustada probleeme.

3.1 Dokumendi uuendamine

Reaalses elus tuleb tarkvara muutmist ja täiendamist tihti ette, selle tagajärjel on vaja muuta dokumentatsiooni. Olenevalt muutusest võib olla tarvis:

1. Muuta olemasolevat dokumentatsiooni
2. Lisada uut dokumentatsiooni
3. Eemaldada olemasolev dokumentatsioon.

Ühtset protseduuri muudatuste ja täienduste juhtimiseks ei ole, see sõltub organisatsiooni suurusest, olemusest ja kultuurist. Tavaliselt algab see protsess muudatuse sooviga, millele järgneb selle hindamine ning heakskiitmine või tagasilükkamine [21]. Kui muudatuse soov kiidetakse heaks on järgmine samm selle rakendamine.

Dokumentatsioonis leidub aegajalt dubleerimist, see tähendab sama loogika on kirjeldatud mitmes kohas. Põhjuseks võib olla äriprotsess, kus ongi ühe loogika kasutamine mitmetes kohtades vajalik. Dubleerimisega tuleb olla eriti ettevaatlik dokumentatsiooni uuendades, et kõikides kohtades oleks sama asi kirjeldatud ühte moodi. Dokumendid, milles on sama osa dubleeritud, kuid kajastatud erinevalt toob endaga kaasa probleeme, kuna siis pole võimalik kindlaks teha, kuidas on õige.

Suurtes tarkvara projektides tekib küsimus, milliseid dokumente kaasajastada, milliseid mitte. Kõige kergem on vastata sellele küsimusele alati, et kõiki dokumente tuleb kaasajastada, kuid kas seda tegelikult vaja on ja kas see tasub ära, kuna tegemist on ajamahuka ja seega kuluka tööga. Scott W. Amber kirjutab ühes oma artiklis [22]

dokumentatsiooni kaasajastamise kohta, et uuendada ainult siis, kui see teeb haiget. Kui dokumentatsioon on aegunud ning tegemist pole kriitilise dokumendiga, siis ei ole selle uuendamine prioriteetne, kuna see ei tee suurt kahju. Lisaks toob Scott W. Amber artiklis välja järgmise heuristika, mille järgi otsustada kas uuendada dokumenti või mitte:

1. Mudeleid tuleb uuendada enne või hiljemalt paralleelselt kui rakendust uuendatakse
2. Dokumentatsioon, mis on osa süsteemist, nagu kasutusjuhend, tuleb uuendada enne süsteemi uuendamist või samal ajal
3. Dokumentatsioon, mille sihtgrupp on klient, peab olema kaasajastatud, kuna see võib tuua endaga kaasa märkimisväärse kahju (nn teeb haiget)

Reaalses elus on dokumentatsioon, mis ei vasta täpselt süsteemile, äärmisel ebameeldiv, kuid tegelikult on kogu dokumentatsiooni uuendamine nii suur ajakulu, et efektiivsus on kogu meeskonnal suurem kui kaasajastatakse, vaid äärmiselt vajalikku dokumentatsiooni. Lõppudelõpuks tahetakse siiski arendada tarkvara, mitte kirjutada dokumente [22].

Kokkuvõtteks on dokumentatsiooni uuendamine äärmiselt riskantne töö, kuna ühelt poolt on tegemist kalli ehk kuluka tööga ja teiselt poolt võib olla tegemist kas väga ebaolulise või vastupidiselt kriitilise tähtsusega tööga, olenevalt dokumendist.

3.2 Versioneerimine

Dokumentatsiooni kaasajastamisega on tihedalt seotud versioneerimise teema, et näidata toimunud muudatusi dokumentides. Versioneerimine annab võimaluse: [23]

1. Vaadata varasemaid dokumendi versioone
2. Võrrelda erinevaid versioone
3. Tagasi pöörata varasemaid versioone
4. Eemaldada varasemaid versioone

Dokumentatsiooni versioneerides on väga oluline anda igale versioonile oma number, mis peab olema unikaalne. Seda numbrit nimetatakse 'Versiooni number' [24].

Versiooni numbri süsteemis kasutatakse ühe komakohaga numbreid suuremate ja väiksemate muudatuste versioneerimiseks. Nummerdamist alustatakse 0.1-st. Väiksemateks muudatusteks nimetatakse näiteks õigekeele parandusi, kontaktandmete muutmist jne. Väikseid muudatusi kajastatakse versiooni numbri kümnendiku suurendamisega. Kui dokumenti muudetakse oluliselt rohkem, kui eelnevalt nimetatud, siis on tegemist suurte muudatustega ning neid kajastatakse versiooni numbri suurendamisega täisarvuks [24]. Vt tabel 1.

<u>Muudatus</u>	
Väiksed muudatused	Dokumendi versioon 0.1 (esimene versioon) Dokumendi versioon 0.2 (teine versioon) Dokumendi versioon 0.3 (kolmas versioon)
Suured muudatused	Dokumendi versioon 1.0 (esimene heakskiidetud versioon) Dokumendi versioon 2.0 (teine heakskiidetud versioon) Dokumendi versioon 3.0 (kolmas heakskiidetud versioon)
Väiksemad muudatused ja suuremad muudatused	Dokumendi versioon 1.0 Dokumendi versioon 1.1 (esimene väiksem muudatus esimeses heakskiidetud versioonis) Dokumendi versioon 2.4 (neljas väiksem muudatus teises heakskiidetud versioonis)

Tabel 1, versioneerimise näide [24]

Versiooni number peab olema dokumendis alati kirjas, et kui inimesed, kes soovivad tutvuda või töötada kindla versiooniga dokumendist saaksid seda kontrollida, milline dokumendi muudatus nende käes parajasti on. Üks levinud võimalus on kasutada dokumendi alguses kontrolltabelit (*ingl. k Document Control Table*), milles kirjeldatakse informatsiooni dokumendi ajaloo kohta. Näiteks autori nimi, dokumendi staatus, versiooni number, heakskiitmise kuupäev [24]. Vt tabel 2.

Kontrolltabel			
Autor	Elen Kivi	Staatus	Heakskiidetud
Versioon	2.3	Kuupäev	12.03.2011
Heakskiidetud:	Aita-Leida Kuusepuud	Kuupäev:	16.03.2011
Asukoht:	http://dokumendiasukoht.ee/dokument		

Tabel 2, dokumendi kontrolltabel [24]

Lisaks võib kasutada eraldi dokumendi ülevaatusetabelit (*ingl.k Documetn Review Table*) dokumentatsiooni versioneerides. Ülevaatusetabelisse kirjutatakse kõik muudatused ning heakskiidetud versiooni kohta tehakse sissekanne kontrolltabelisse [24]. Vt tabelit 3.

Ülevaatusetabel			
Versioon	Muudatus	Autor	Kuupäev
2.1	Õigekeelsus parandused	E. Kivi	17.03.2011
2.2	Privileegide muudatused	E. Kivi	23.03.2011
2.3	Kontaktandmete muutmine	E. Kivi	30.03.2011
Suletud	Asendatud versiooniga 3.0	E. Kivi	01.04.2011

Tabel 3, dokumendi ülevaatusetabel [24]

Lisaks eelnevalt nimetatule on võimalik elektrooniliste dokumentide puhul salvestada versiooni number ka faili nimesse. Näiteks Dokument_V0.1 või Kasutuslugu_V2.1

4 Uurimuse kirjeldus

4.1 Küsitluse ettevalmistamine ja läbiviimine

Uurimuse tegemiseks viidi läbi rida küsitlusi. Uurimuses osalesid kahe erineva osapoolle esindajad: kliendid ehk tarkvara tellijad ja arendajad ehk tarkvara tegijad.

Mõlemad osapooled vastasid küsitlusele, mis oli veidi erinev. Klientidele mõeldud küsimustik oli koostatud vaadatuna dokumentatsiooni kliendi vaatenurgast ja arendaja küsimustik oli vastupidine ja vaatas dokumentatsiooni arendaja vaatenurgast. Küsimustik oli lühike, et sellele vastamiseks ei kuluks kaua aega.

Uurimuses osalejate leidmiseks kasutati interneti, et leida erinevaid Eestis tegutsevaid tarkvaraarendusega tegelevaid ettevõtteid ning organisatsioone, kes on tellinud infosüsteeme. Esialgu valiti juhuslikkuse alusel välja 20 arendajat ja 20 klienti, kuid neist vastasid küsimustikule vaid mõned. Seejärel suurendati valimit ning otsiti arendajaid interneti kataloogist neti.ee ja saadeti kõigile seal registreeritud tarkvara arendusega tegelevatele ettevõtetele küsimustik. Eesti Riigihangete registrist leiti infot nii klientide kui arendajate kohta. Lisaks uuriti tarkvaraarenduse teenust pakkuvate firmade kodulehti ning leiti viiteid ka nende klientidele. Kokku leiti 100 klienti ja 100 arendajat, kes on potentsiaalsed vastajad. Kõigile neile saadeti e-kiri, koos küsitluse lingiga. Mõningate klientide poolt tuli vastuseid, et nemad ei ole kokku puutunud tarkvaraarendusega ja ei ole pädevad vastama antud küsimustikule. Põhjuseid oli erinevaid, mõnedes organisatsioonides oli töölt juba lahkunud inimene, kes tegeles tarkvaarendusega, teistes jälle oli see teenus sisse ostetud. Lisaks saatsid mõned inimesed veel täpsustusi oma vastuste kohta, kuna leidsid, et on oluline lisada nüansse, mis nende organisatsiooni puudutab.

Kokku vastas küsitlusele 41 ettevõtet, neist 22 olid arendajad ja 19 kliendid. Klientide seas oli suurem osa vastanutest riigiasutused, kuid mõned ettevõtted kuulusid ka erasektoris. Küsitlusele vastanud arendusega tegelevatest ettevõtetest olid kõik erasektori omad.

Analüüs jagunes järgmistesse gruppidesse:

1. Küsimused, mis puudutasid infosüsteemi eripärasid

2. Küsimus koostatava dokumentatsiooni kohta
3. Küsimus dokumentatsiooni kohta, mis on vajalik või ebavajalik
4. Küsimus, milliseid dokumente kaasajastatakse
5. Küsimused, mis olid arendajale ja kliendile erinevad
 - a. Arendajatelt küsiti, miks uuendatakse neid dokumente
 - b. Klientidelt küsiti, kuidas hindate dokumentatsiooni kaasajastatust

4.2 Arendaja küsimustiku tulemused

Antud peatükis vaadatakse küsitluse vastuseid arendaja vaatenurgast.

Esimene küsimus, mis oli arendajatele mõeldud, puudutas nende kliente:

Kas Teie klient on riiklik asutus?

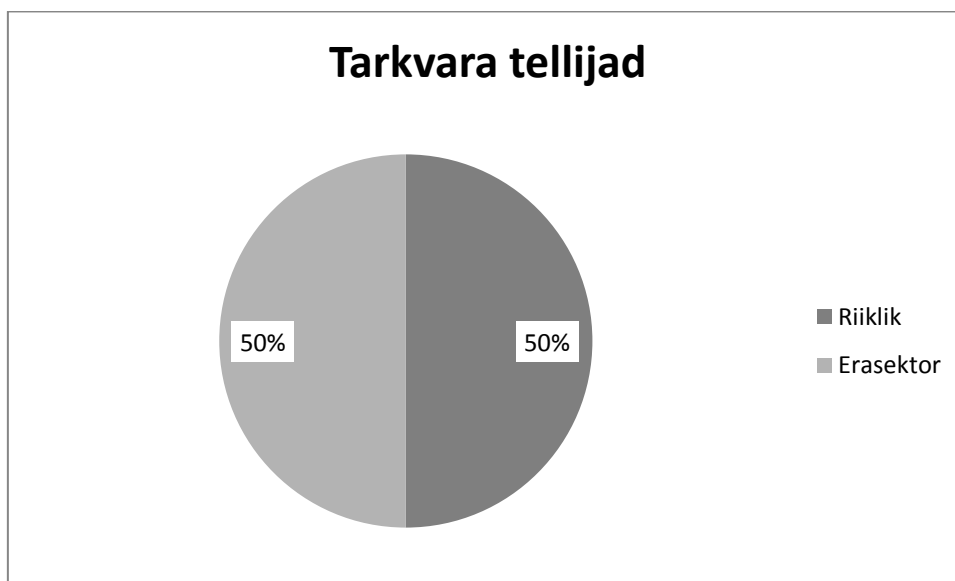


Diagramm 1, Tarkvara tellija

Vastused jagunesid täpselt pooleks, 22-st vastanud tarkvaararenduse firmast pooled vastasid, et nende klient on riiklik asutus. Seega saab edaspidi vaadata, kuidas mõjutab dokumentatsiooni see, kas tarkvara tellija on riiklik asutus või kas üldse see mõjutab.

Järgmine küsimus puudutas infosüsteemi kasutamist:

Kas tegemist on avaliku rakendustega?



Diagramm 2, Avalik või kinnine rakendus

Rakendust, mida ei saa kasutada kõik inimesed ja mis on mõeldud mõne organisatsiooni siseseks kasutamiseks, nimetatakse antud töös kinniseks rakenduseks ehk suletud kasutajate ringile mõeldud süsteemiks. Diagrammilt 2 on näha, et enamus küsimustikule vastanud tarkvaraarendajaid töötab kinnise rakendusega. Ainult 36% vastanutest töötab avalikkusele mõeldud rakendusega ja 64% suletud kasutajate ringile mõeldud rakendusega. Põhinedes kahele eelnevale küsimusele on võimalik jagada kõik vastanud arendajad nelja rühma:

1. Tellija on riiklik asutus ja arendatakse avalikku rakendust
2. Tellija on riiklik asutus ja arendatakse kinnist rakendust
3. Tellija kuulub erasektoris ja arendatakse avalikku rakendust
4. Tellija kuulub erasektoris ja arendatakse kinnist rakendust

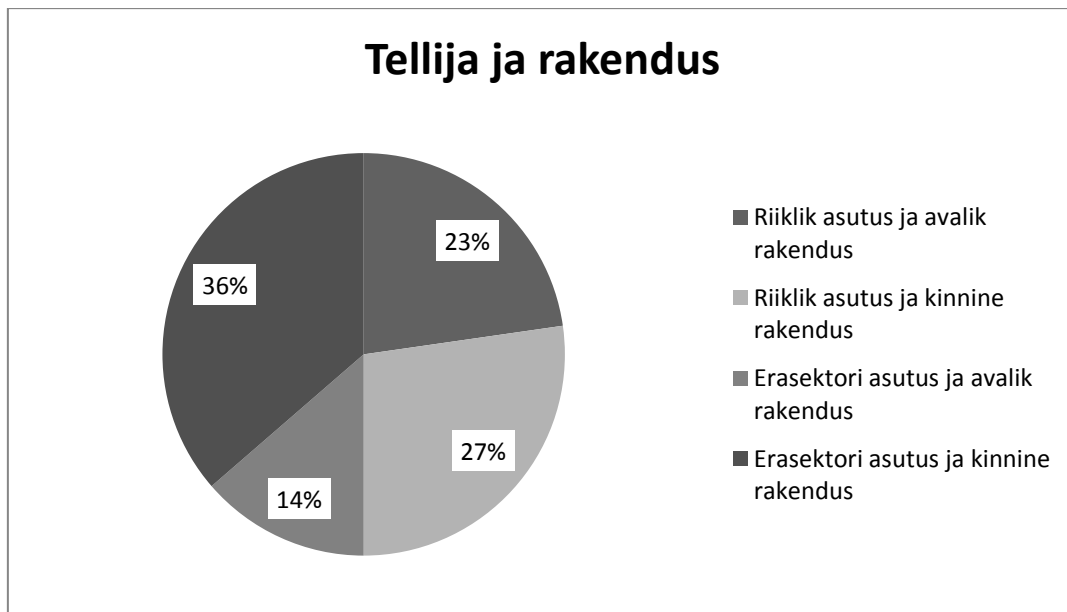


Diagramm 3, Tellija ja rakendus

Diagrammilt 3 on näha, et kõige suurem osa vastanud ettevõtetest kuuluvad erasektoris, kes arendavad suletud kasutajate ringile.

Kasutajate arvu kohta oli järgmine küsimus:

Kui palju kasutajaid on rakendusel?

Antud küsimuse eesmärgiks oli uurida, kuidas mõjutab IT süsteemi dokumentatsiooni selle kasutajate arv. Antud küsimusele jätsid küsimustikus erinevatel põhjustel vastamata 3 ettevõtet, seega põhinevad järgmised arvutused 19 süsteemi kasutajate arvul. Kasutajate arvu aritmeetiline keskmine on 11 400 kasutajat. Kõige rohkem kasutajaid on 100 000 ja kõige väiksema arvu kasutajatega süsteemil on, vaid üks kasutaja. Kasutajate arvu mediaan on aga 100, mood 100, standard hälve on 31 270.

Järgmine küsimus oli koostatava dokumentatsiooni kohta:

Milliseid dokumente koostate?

Vastajatele anti ette nimekiri dokumentidest ja võimalus lisada ise ka teisi dokumente ning sooviti teada, milliseid dokumente tarkvaararendaja koostab. Valikus olid järgmised dokumentide liigid:

1. Projektiplaan
2. Riskianalüüs
3. Nõuete nimekiri
4. Kasutuslood

5. Tehnilised spetsifikatsioonid
6. Andmemudel
7. Arhitektuuri dokument
8. Disaini dokument
9. Testjuhtumid
10. Testplaan
11. Taasteplaan
12. Paigaldusjuhend
13. Kasutusjuhend

Uurimuse tulemustest on näha, et kõige rohkem, 16 ettevõtet kirjutab kasutuslugusid ja 15 tehnilisi spetsifikatsioone ehk analüüsi etapiga seotud dokumentatsiooni. Lisaks eelnevalt nimetatule märkisid vastanud arendajatest 13, et koostavad projektiplaani, kasutusjuhendi ja paigaldusjuhendi. Kõige vähem, vaid 4 ettevõtet, koostab taasteplaani. Samuti koostatakse märkimisväärselt vähe riskianalüüsi, 6 ettevõtet 22-st vastanust märkisid, et koostavad antud dokumendi.

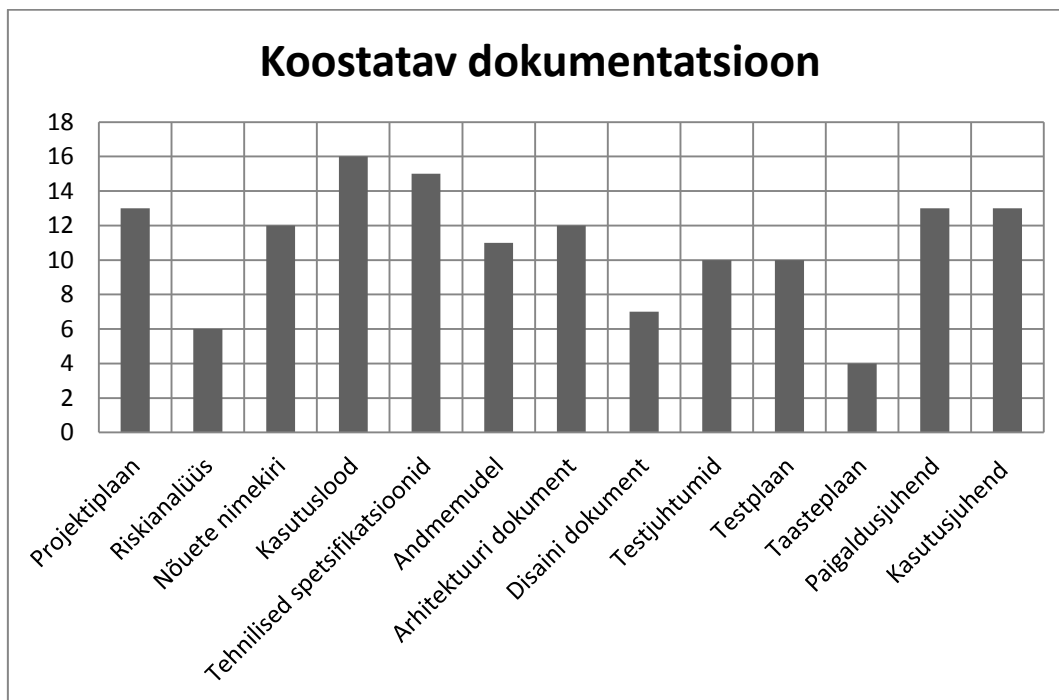


Diagramm 4, Koostatav dokumentatsioon

Lisaks eelnevalt nimetatud dokumendi liikide valikule, oli vastajal võimalik valida vastuse variant muu ning lisada teisi dokumendi liike. Üks arendaja kasutas antud võimalust ning lisas koostatava dokumentatsiooni hulka ka prototüübi.

Järgmine küsimus sarnanes veidi eelmise küsimusega, kuid küsiti:

Milliseid dokumente peate vajalikeks?

Küsimuse eesmärk oli välja selgitada, milliseid dokumente peab arendaja vajalikuks, kuna tihti võib määrata leping või jälgitav standard või kliendi soov koostatava dokumentatsiooni. Küsimuses oli sarnaselt eelmisele küsimusele ette antud valik dokumendi liike ja võimalus lisada ka teisi. Uurimuse tulemusena selgus, et kõige vajalikemaks peetakse kasutuslugusid, tehnilisi spetsifikatsioone ja arhitektuuri dokumenti, 15 arendajat 22-st vastanust arvasid nii. 14 arendajat pidasid vajalikuks ka paigaldusjuhendit ja 13 kasutusjuhendit. Teisi dokumente pidasid vajalikuks vähem vastanud ettevõtetest. Kõige ebavajalikum dokument tarkvaraarendajate arvates on taasteplaan, mida pidasid vajalikuks vaid 5 vastanut. Riskianalüüsi dokument ei ole uurimuse tulemuste järgi samuti väga vajalik, ainult 7 arendajat valisid selle vastuse variandi. Diagrammil 5 on näha dokumendi liikide kaupa, kui vajalikuks Eesti tarkvaraarenduse ettevõtted neid peavad.

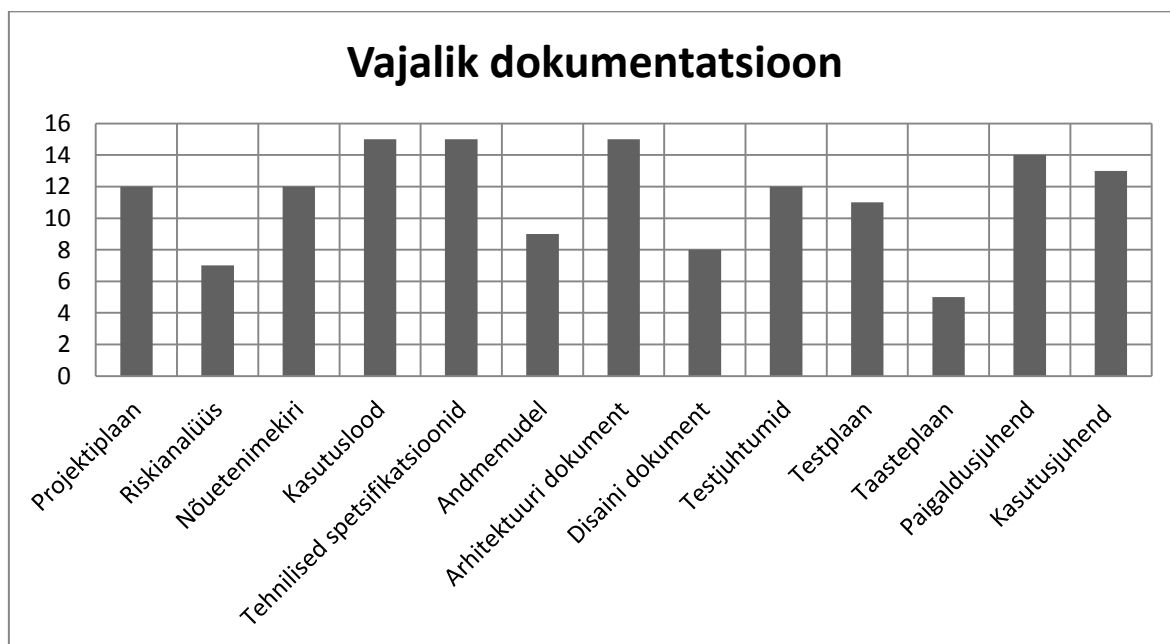


Diagramm 5, Vajalik dokumentatsioon

Järgmine küsimus puudutas dokumentatsiooni kaasajastamist:

Milliseid dokumente kaasajastate pidevalt (up-to-date)?

Küsimuse eesmärk oli välja selgitada, milliseid dokumendi liike tarkvaraarendajad kaasajastavad ja milliseid mitte. Sarnaselt kahele eelmisele küsimusele oli ka selle küsimuse juures vastajale ette antud valik dokumendi liikidest. Tulemused on võrreldes eelmiste küsimustega palju väiksemad, see tähendab dokumentatsiooni hulk, mida kaasajastatakse, ei ole nii suur kui oleks võinud arvata, vaadates koostatava ja vajaliku dokumentatsiooni küsimuste tulemusi. Kõige tihedamini kaasajastatakse kasutusjuhendit, nii vastasid 10 tarkvaraarendajat 22-st. Projektiplaani, kasutuslugusid, tehnilisi spetsifikatsioone ja paigaldusjuhendit uuendab 7 ettevõtet. Kõige väiksem uuendamise osakaal on disaini dokumendil, mida kaasajastab vaid 2 arendajat. Diagrammil 6 on näha kõigi dokumendi liikide kohta, kui paljud vastanud tarkvaraarendajatest neid kaasajastavad.

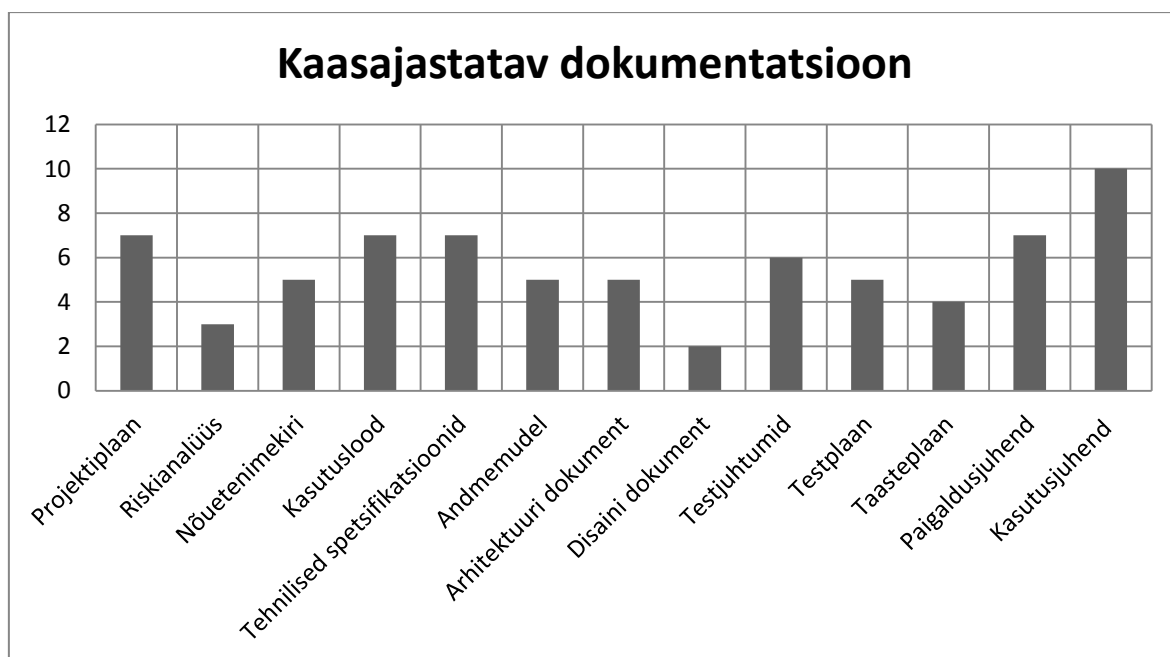


Diagramm 6, Kaasajastatav dokumentatsioon

Viimane küsimus arendajatele oli samuti dokumentatsiooni kaasajastamise kohta:

Miks kaasajastate neid dokumente?

Antud küsimuse vastust oodati vabas vormis ja seetõttu on tulemused väga erinevad. Vastustes esitati järgmisi põhjendusi:

1. Sest nende dokumentide alused muutuvad
2. Edasise arenduse teeb lihtsamaks
3. Need vananevad kõige kiiremini, töö efektiivsuse huvides on vaja nad aktuaalsed hoida
4. Haldusmurede lahenduseks
5. Meie eesmärgiks on pakkuda tarkvara, mida kasutaja kasutada oskab, seetõttu on vajalik ka uuendatud kasutusjuhend (Märkus: antus ettevõtte märkis, et uuendavad, vaid kasutusjuhendit)
6. Et jälgida projekti seisu ja muudatusi
7. Nii enda kui haldajate/kasutajate jaoks - protsessi osa
8. Et oleks endal ka kiire ülevaade süsteemist
9. Vajalik arenduse tegemiseks ja projekti edukaks lõpetamiseks
10. Rakendusse tuleb reeglina täiendusi kogu arendusprotsessi vältel
11. Rakenduse loomulik areng
12. Et programmist tervikülevaadet omada

Kokkuvõttes on dokumentatsiooni kaasajastamisel tarkvaraarendajatel üsna sarnased eesmärgid.

4.3 Kliendi küsimustiku tulemused

Antud peatükis vaadatakse küsitluse vastuseid kliendi vaatenurgast.

Esimene küsimus tarkvara tellija küsimustikus puudutas organisatsiooni kuuluvust:

Kas Teie organisatsioon on riiklik asutus?

Nagu eelnevalt nimetatud oli suurem osa vastanud klientidest riiklikud asutused ning mõned üksikud kuulusid erasektoris. Diagrammilt 7 on näha, et 79% ehk 19 vastanud klienti kuulub riiklikusse sektorisse ja 21% erasektoris.

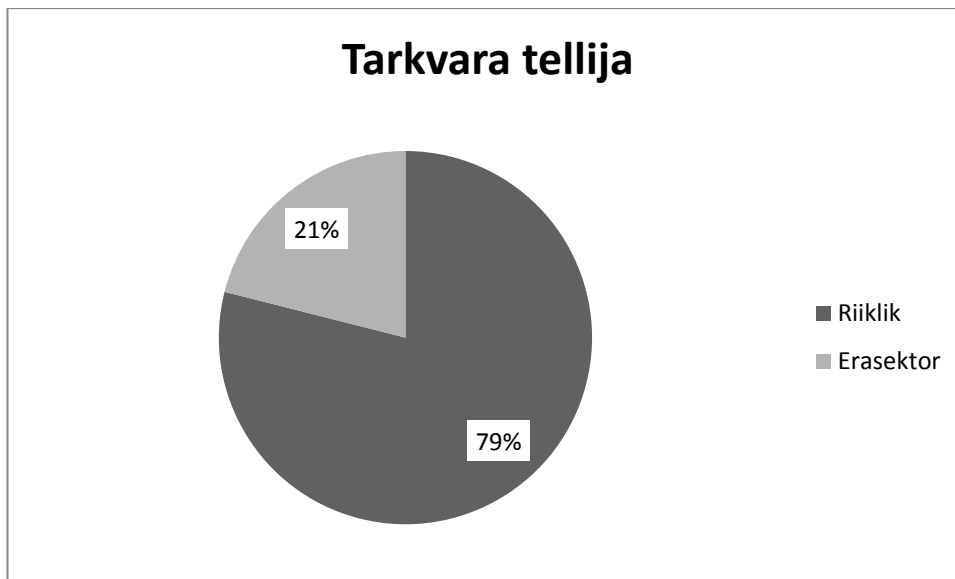


Diagramm 7, Tarkvara tellija

Teine küsimus, mis oli klientidele mõeldud küsimustikus puudutas süsteemi:

Kas tegemist on avaliku rakendusega?

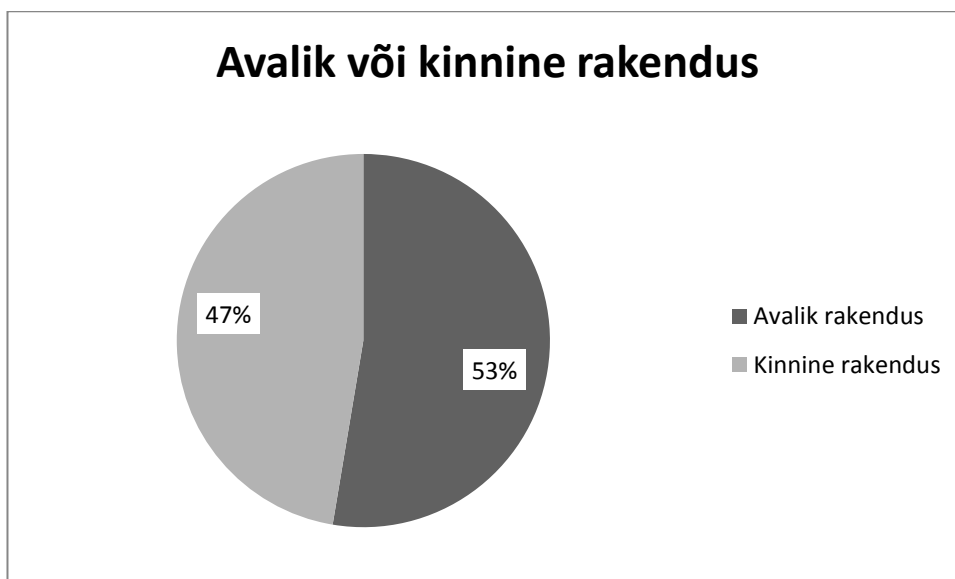


Diagramm 8, Avalik või kinnine rakendus

Diagrammilt 8 on näha, et 53% vastanud klientidest tellib avaliku rakenduse ning 47% kinnise rakenduse, see tähendab, et 19-st vastanud ettevõttest 10 tegeleb avaliku ja 9 kinnise rakendusega. Sarnaselt arendaja küsimustiku vastuste põhiselt on ka võimalik klientide vastuste järgi jaotada kõik vastanud nelja rühma:

1. Tellija on riiklik asutus ja arendatakse avalikku rakendust

2. Tellija on riiklik asutus ja arendatakse kinnist rakendust
3. Tellija kuulub erasektoris ja arendatakse avalikku rakendust
4. Tellija kuulub erasektoris ja arendatakse kinnist rakendust

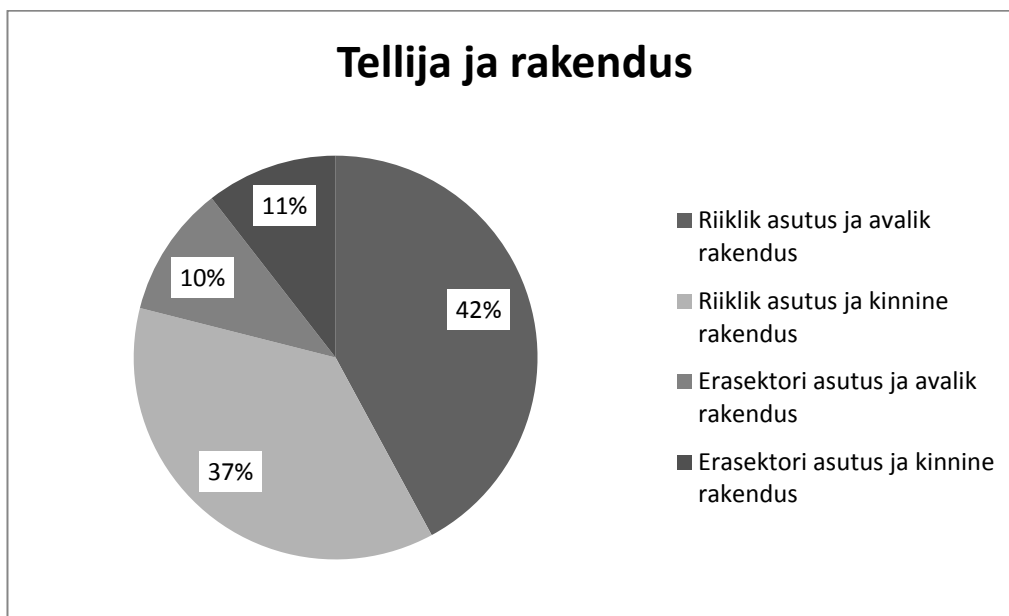


Diagramm 9, Tellija ja rakendus

Vastavalt jaotusele, kuulub kõige suurem osa tellijatest, 42%, gruppi „Riiklik asutus ja avalik rakendus“. Kõige vähem on aga erasektori organisatsioone, kelle rakendus on avalik.

Kolmas küsimus küsis rakenduse kasutajate arvu kohta:

Kui palju on rakendusel kasutajaid?

Antud küsimusele jätsid küsimustikus vastamata 3 ettevõtet, seega põhinevad järgmised tulemused 16 kliendi vastustel. Kasutajate arvu aritmeetiline keskmine on vastanud tarkvara tellijate süsteemides 2 300 kasutajat. Kõige rohkem kasutajaid on 20 000 ja kõige väiksema aru kasutajatega süsteemil on 20 kasutajat. Kasutajate arvu mediaan on 200, mood 200, standard hälve 5 254.

Järgmine küsimus puudutas süsteemi arendamise aega:

Kui kaua on kestnud rakenduse arendamine?

Küsimuse eesmärk oli välja selgitada kui kaua on kestnud infosüsteemi arendus, et näha, kas see mõjutab dokumentatsiooni. Küsimusele jätsid vastamata 3 vastajat ja seetõttu põhineb järgmine analüüs 16 tarkvara telliva ettevõtte vastustel. Küsimuse vastustest selgus, et kõige pikema arendamise ajaga rakendust on arendatud 84 kuud ehk 7 aastat. Kõige lühem arenduse periood on 3 kuud. Keskmise arenduse aeg on vastanud tarkvara tellijate rakendustel 42 kuud ehk 3 ja pool aastat. Mediaan on 33 kuud ehk veidi alla 3 aasta ning mood on 36 kuud ehk täpselt 3 aastat. Standard hälve on 30 kuud.

Järgmine küsimus oli IT süsteemi äriloogika keerukuse kohta:

Kuidas hindate rakenduse äriloogikat?

Küsitluses oli ette antud neli valikuvarianti:

1. Väga keeruline
2. Keeruline
3. Lihtne
4. Väga lihtne

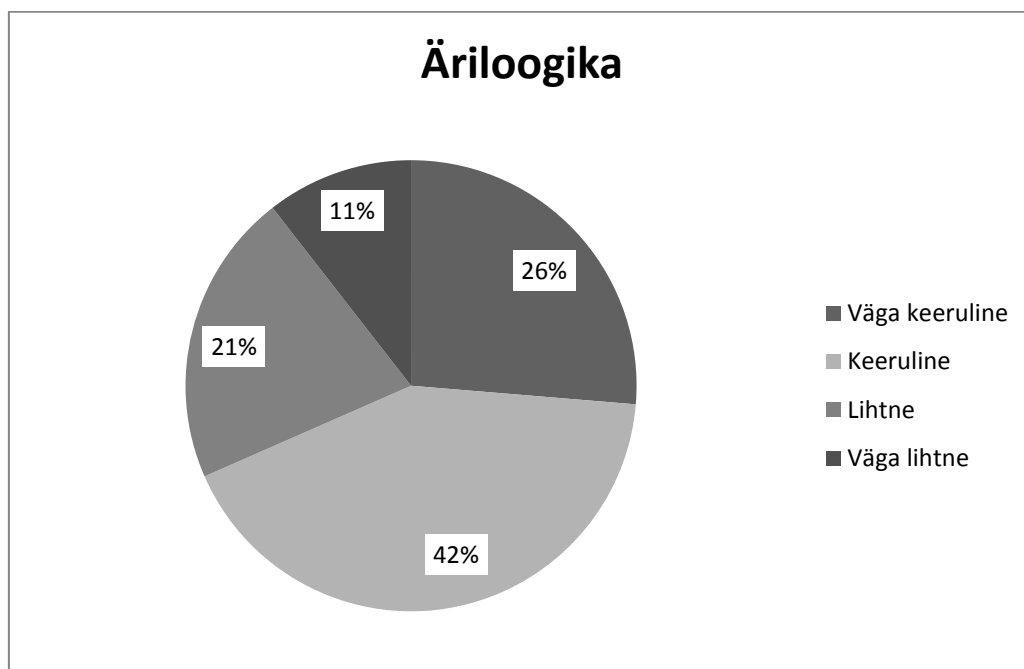


Diagramm 10, IT süsteemi äriloogika

Diagramm 10 annab ülevaate vastuste jagunemisest. Kõige suurem osa vastanutest hindas süsteemi äriloogikat keeruliseks ning kõige väiksem osa vastanutest väitis oma tarkvara äriloogikat olevat väga lihtsana. Lihtsate ja väga keeruliste süsteemide osakaal oli peaaegu võrdne, vastavalt 21% ja 26%.

Järgnevad küsimused puudutasid dokumentatsiooni:

Milliseid dokumente koostab arendaja?

Samad vastuse variandid, mis olid arendajatele mõeldud küsimustikus, anti ette ka klientidele.

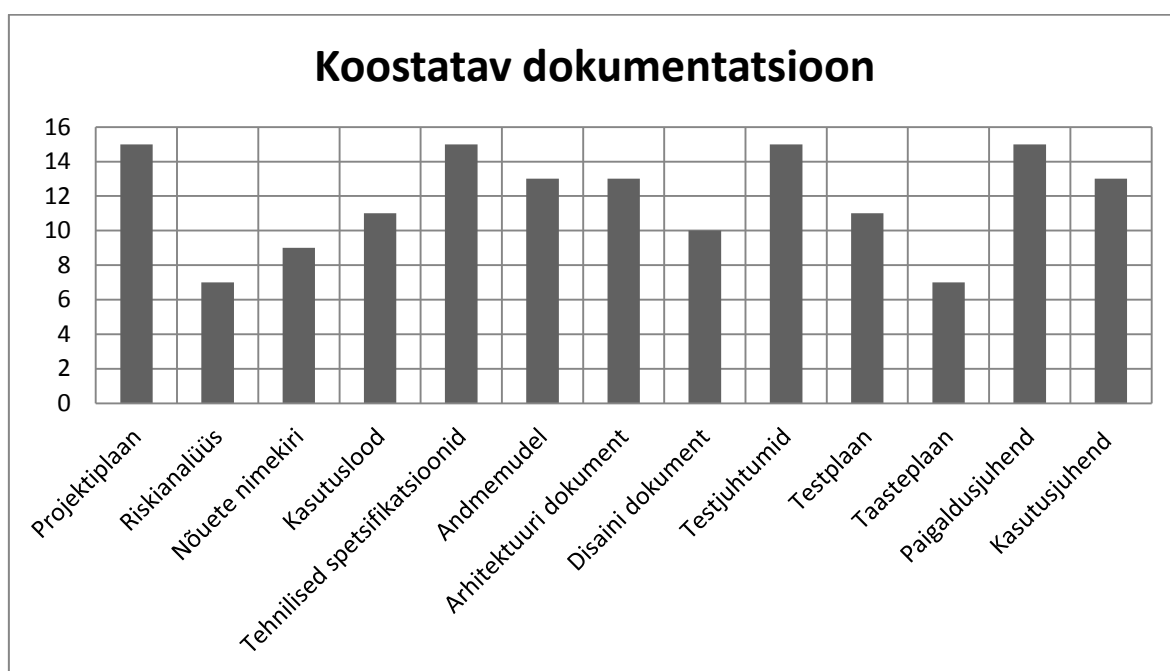


Diagramm 11, Koostatav dokumentatsioon

Diagramm 11 annab ülevaate klientide vastustest, milliseid dokumente koostatakse. Kõige rohkem koostatakse projektiplaani, tehnilisi spetsifikatsioone, testjuhtumeid ja paigaldusjuhendit, nii vastasid 15 ettevõtet 19-st vastanust. Kõige vähem valiti vastuse variantide hulgast riskianalüüsi ja taasteplaani dokumente, see tähendab 7 tellijat saavad need dokumendid arendajalt.

Järgmine küsimus oli tellijate küsimustikus veidi erinev arendajate omast:

Millised dokumendid on neist ebavajalikud?

Küsimuse eesmärk oli uurida, kas leidub dokumente, mida arendaja koostab, kuid mille vajalikust ei oska klient hinnata või pole see lihtsalt tema jaoks vajalik. Küsimuses selgus, et ebavajaliku dokumentatsiooni on kliendi jaoks väga vähe. Ainult 2 vastajat oskas tuua välja mõned dokumendid, mis on nende arvates ebavajalikud. Esimene neist nimetas, et andmemudel, paigaldusjuhend ja taasteplaan on ebaolulised. Teine tõi välja kasutusloo ja disainidokumendi.

Järgmine küsimus puudutas puuduolevat dokumentatsiooni:

Milliseid dokumente peaks veel arendaja koostama?

Küsimuse eesmärk oli välja selgitada, kas kliendi jaoks leidub dokumentatsiooni, mida arendaja ei koosta, kuid klient peab seda vajalikuks. Sarnaselt eelmise küsimusega oli ka sellel üsna vähe vastajaid. 19 küsitlusele vastanud tellijast tõi ainult 7 välja puuduvaid dokumente.

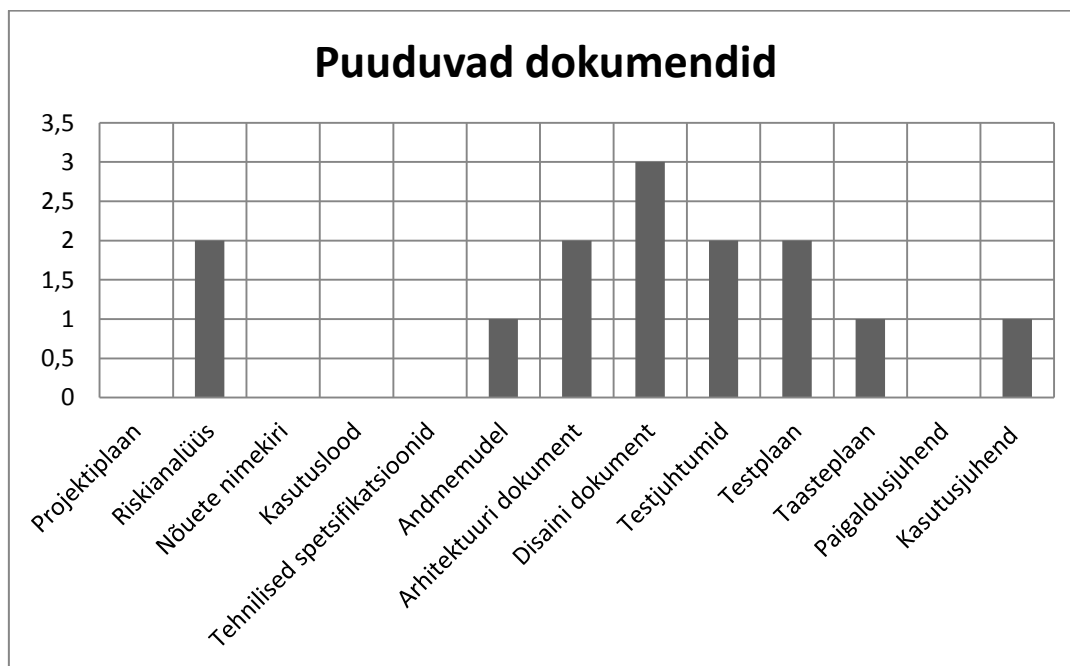


Diagramm 12, Puuduvad dokumendid

Diagrammilt 12 on näha, et kõige suurem puudujääk on disaini dokumendi osas. Teisi dokumendi liike toodi vähem välja.

Viimane küsimus klientidele puudutas dokumentatsiooni kaasajastamist:

Kuidas hindate dokumentide kaasajastamist ehk dokumentatsiooni uuendamist?

Vastajatele anti ette kolm valikuvarianti:

1. Piisav, vajalik dokumentatsioon on kaasajastatud
2. Keskpärane, on puudujääke
3. Kasin, palju vajalikku dokumentatsiooni on aegunud

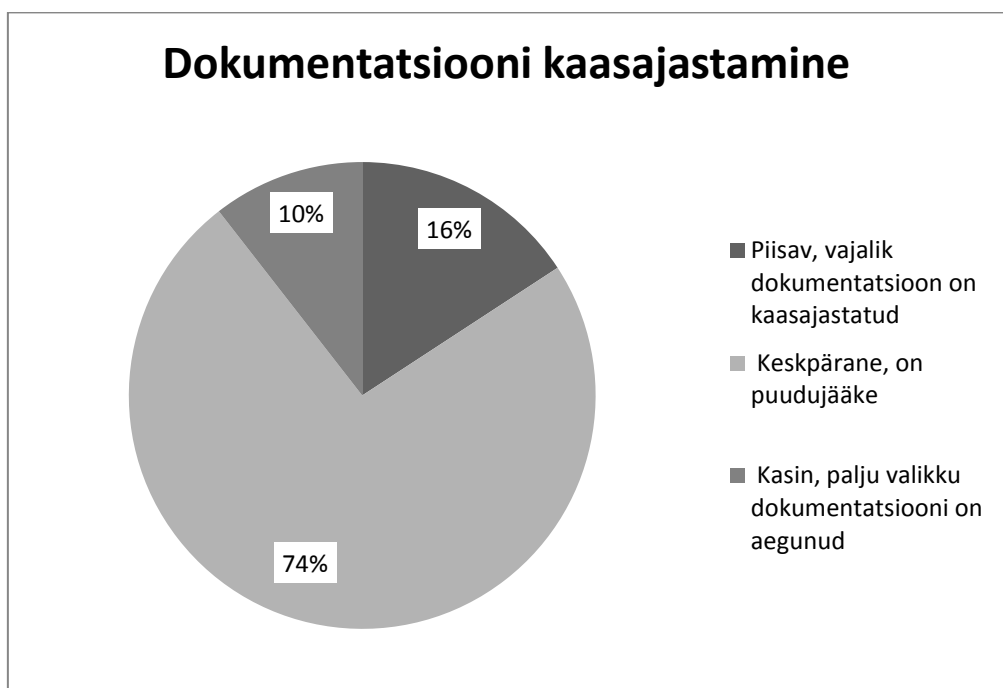


Diagramm 13, Dokumentatsiooni kaasajastamine

Suur enamus tellijatest leiab, et dokumentatsiooni kaasajastamine on keskpärane, leidub puudujääke. 16% vastanutest hindas dokumentatsiooni kaasajastamist piisavaks, see tähendab vajalik dokumentatsioon on kaasajastatud. Ainult 10% tellijatest leidis, et dokumentide uuendamine arendaja poolt on kasin ja palju vajalikku dokumentatsiooni on aegunud.

5 Tulemuste analüüs

Töö eesmärgiks oli uurida, millist dokumentatsiooni koostab või ei koosta arendaja ja millist dokumentatsiooni peab vajalikuks tellija ning milliseid dokumente kaasajastab arendaja ja kuidas hindab klient uuendamise tulemust.

Uurimuse käigus selgus, et välja pakutud 13 erinevat dokumendiliiki koostatakse kõiki, kuigi erinevates organisatsioonides koostatakse erinevaid alamosasid välja pakutud dokumentidest.

Uurimistöö esimene hüpotees oli, et tellija praktiline vajadus dokumentatsiooni järgi erineb sellest, millist dokumentatsiooni arendaja teeb. Kui võrrelda kahte diagrammi, ühte dokumentatsiooni kohta, mida koostab arendaja, diagramm 4, ja teist tellija poolt vaadatuna, diagramm 11, siis selgub, et nende vahel on erinevused. Järgnevalt tuuakse välja kolm kõige suurema erinevusega dokumendi liiki, see tähendab, kas arendajad koostavad suuremal või vähemal määral, kui kliendid soovivad, vt tabel 7.

Nõuete nimekiri on dokument, mida arendajad on märkinud 13 korral, kuid kliendid 7 juhul, et seda dokumenti koostatakse. Protsentuaalselt on see 59% vastanud arendajatest väitsid, et koostavad nõuete nimekirja ning ainult 37% klientidest ütlesid, et koostatavate dokumentide hulka kuulub nõuete nimekiri. Pääaegu sama suur erinevus on ka kasutuslooga, kus 16 arendajat ja 10 klienti vastasid, et seda dokumenti koostatakse. Protsentidest väljendades teeb see 73% vastanud arendajatest ja 53% vastanud klientidest. Järelikult on tegemist dokumentidega, mida arendajad koostavad, kuid kliendid ei telli või ei kasuta.

Vastupidine suhe on aga testjuhtumitega, mida koostab 11 arendajat vastanud 22-st ja 14 klienti 19-st märkisid, et testjuhtumeid koostatakse. Kui vaadata protsente on erinevus 24%. Testjuhtumite puhul võib olla põhjustatud erinevus sellest, et osad tellijad koostavad testjuhtumid ise ning arendajad ei koosta neid, vaid testivad ilma testjuhtumiteta. Järelikult on tegemist dokumendiga, mida kliendid vajavad enam kui arendajad.

Dokumentatsiooni koostavad		
	Arendaja	Klient
Projektiplaan	64%	74%
Riskianalüüs	27%	32%
Nõuete nimekiri	59%	37%
Kasutuslood	73%	53%
Tehnilised spetsifikatsioonid	73%	74%
Andmemudel	50%	63%
Arhitektuuri dokument	59%	63%
Disaini dokument	36%	47%
Testjuhtumid	50%	74%
Testplaan	50%	53%
Taasteplaan	18%	32%
Paigaldusjuhend	64%	74%
Kasutusjuhend	64%	58%

Tabel 4, Arendaja ja tellija koostatav dokumentatsioon protsentuaalselt

Uurimuse eelselt püstitatud teine hüpotees oli, et arendaja poolt koostatud oluline dokumentatsioon on aegunud ja seega ei vasta süsteemile. Hüpoteesi tõestamiseks uuriti tellijatelt, milline dokumentatsioon on ebavajalik, et saada teada, milliste dokumentide kaasajastamist oodatakse. Kuna sellel küsimusele vastas väga vähe kliente tähendab see, et suurem osa peab kõiki koostatavaid dokumente oluliseks. Sarnaselt uuriti arendajatelt, milline dokumentatsioon on vajalik ja mida uuendatakse. Üllatuslikult peavad tarkvaraarendajad palju suuremat hulka dokumentatsiooni vajalikuks kui nad kaasajastavad. Diagrammilt 14 on näha, et nõuete nimekirja peab 55% arendajatest oluliseks, kuid sellest hoolimata, kaasajastab seda, vaid 23% vastanutest. Sarnaselt nõuete nimekirjaga on kasutuslugudega, tehniliste spetsifikatsioonidega, arhitektuuri dokumendiga, disaini dokumendiga, testjuhtumitega, testplaaniga ja paigaldusjuhendiga. Sellest võib järeldada, et reaalselt neid dokumente siiski ei kasutata, kuna kaasajastamata dokument ei edasta korrektset infot ja on seetõttu kasutu. Selle põhjuseks võib olla ajapuudus või finants pool, kuna dokumentatsiooni uuendamine on äärmiselt kulukas töö.

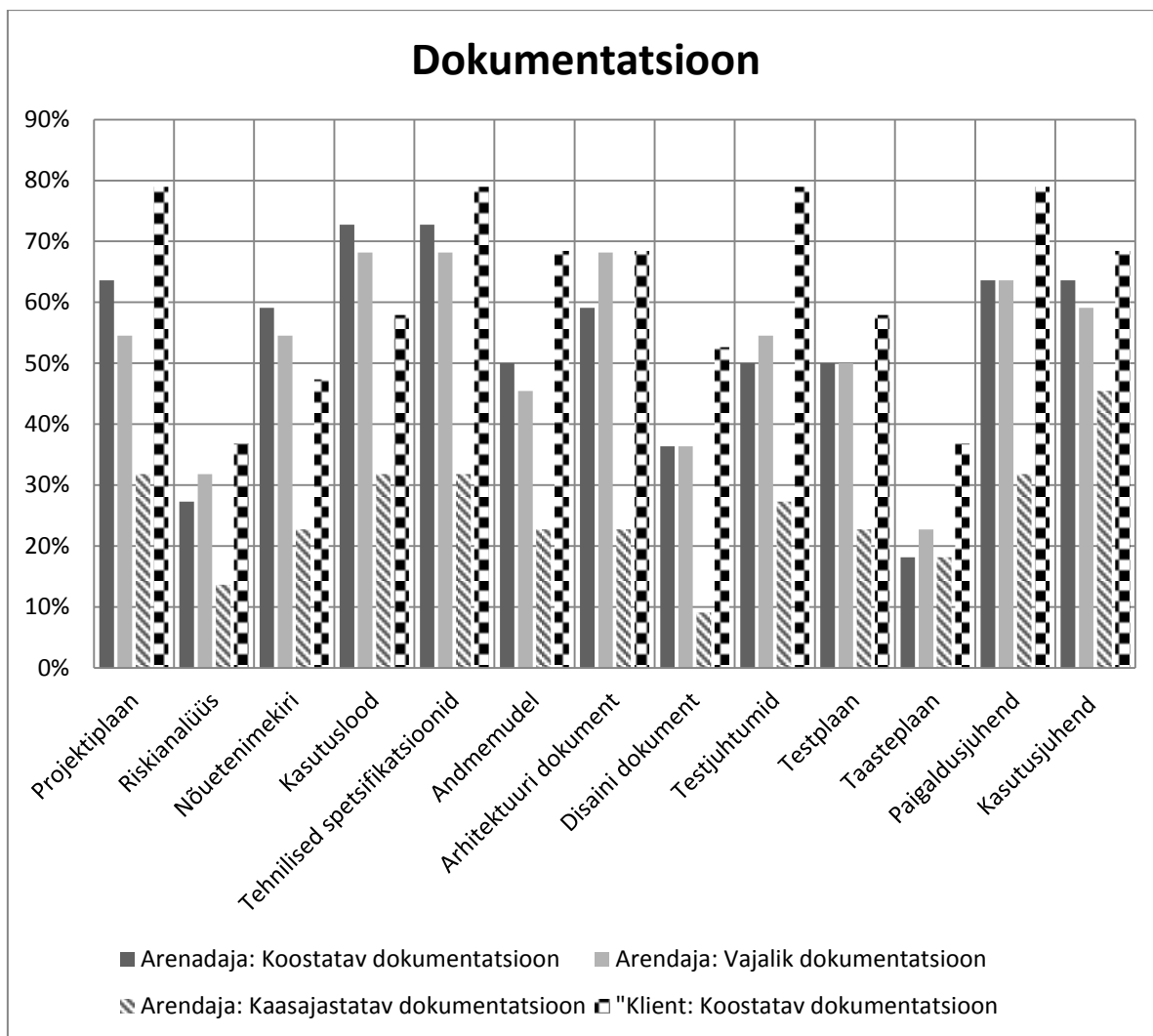


Diagramm 14, Dokumentatsiooni kokkuvõtte

Kui vaadata diagrammi 14 kliendi poolelt, ehk võrrelda tarkvara tellija poolt märgitud koostatavat dokumentatsiooni ja arendaja poolt kaasajastatavaid dokumente, siis ka siin tulevad välja suured erinevused. Vaatame kõige suurema protsentuaalse erinevusega dokumendi liike, ehk mida kõige suurem osa klientidest peab vajalikuks, kuid kõige väiksem osa arendajatest kaasajastab.

Vastanud tarkvara tellijatest 50% vastasid, et testjuhtumid koostatakse, kuid ainult 27% arendajatest märkis, et kaasajastavad seda dokumenti. Väga suur lõhe oli ka projektiplaani, tehniliste spetsifikatsioonide ja paigaldusjuhendi koostamisel. Kõigi nende dokumendi liikide koostamise protsent oli väga suur klientidele tehtud küsitluse tulemusena ning kuna ebavajalikuks ei nimetatud dokumente, siis on need dokumendid tellijale olulised. Teiselt poolt arendajatele tehtud küsitluses vastas väga väike osa, et kaasajastavad neid dokumendi

liike. Uurimuses selgus ka, et kõige parem seis on kasutusjuhendiga, mida 68% klientidest pidasid vajalikuks ja 45% arendajatest märkis, et kaasajastavad seda.

Lisaks kaasajastamise küsimusele dokumentide liikide kaupa, annab uurimusele lisaväärtus küsimus dokumentide kaasajastamise rahulolu koht klientidelt. Küsimuse tulemused on näha diagrammil 13, mille keskmine on 2,0 3-st. Tulemus on keskpärane, mis tähendab, et osa dokumentatsiooni on siiski kaasajastamata.

Hüpotees on seega tõestatud, kuna leidub dokumente, mis on tellija seisukohast olulised, kuid väike osa tarkvaraarendajatest kaasajastab neid.

Kolmas püstitatud hüpotees oli süsteemi eripärade kohta. Eeldati, et süsteemi arendamise kestvus ja keerukus, avalik või kinnine rakendus jne mõjutavad dokumentatsiooni koostamise hulka. Järgnevalt vaatame erinevate süsteemi eripärade mõju dokumentatsioonile.

Küsimustikus oli küsimus, kas tegemist on riikliku asutusega. Hüpoteesi kohaselt eeldati, et see, kas tegemist on riikliku või avaliku asutusega võiks mõjutada. Arendaja küsimustiku tulemustena selgub, et enamuse dokumendi liikide osas ei ole suuri erinevusi, kuid projektiplaani ja paigaldusjuhendit koostatakse riikliku sektor ettevõtetes enam.

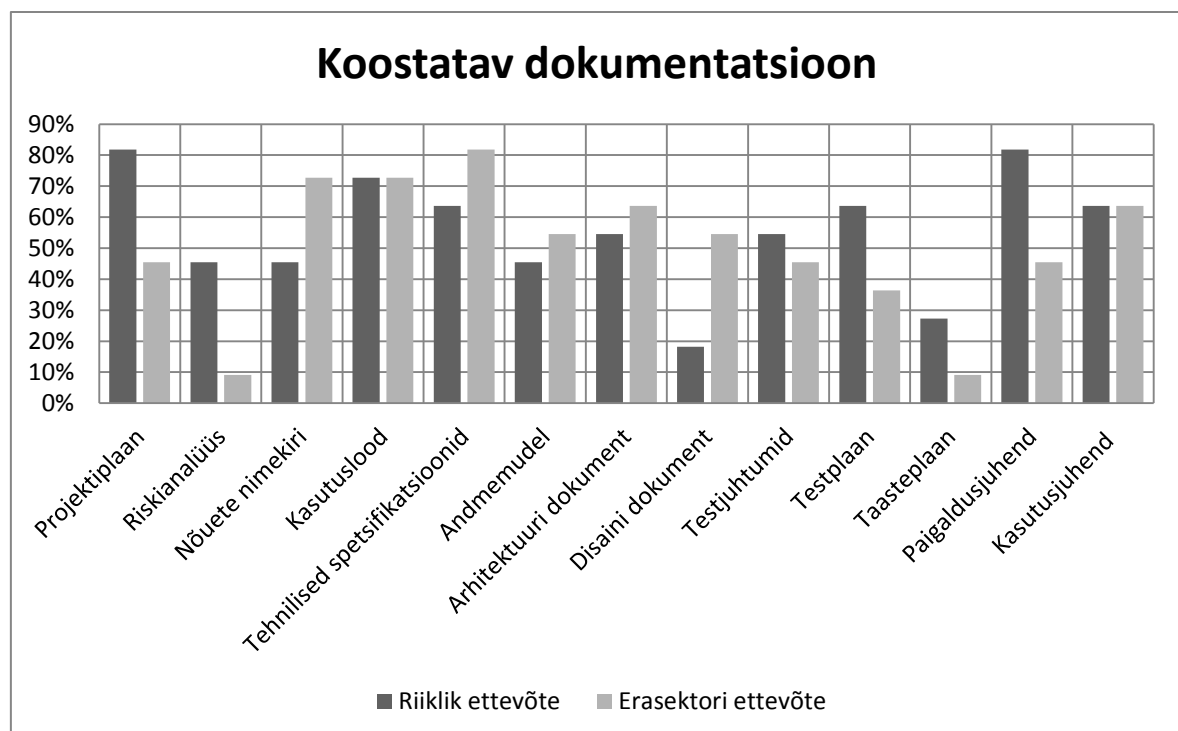


Diagramm 15, Koostatav dokumentatsioon riikliku ja erasektori ettevõtetes

Tellijate küsitlusest ei tule antud erinevus välja, kuna klientidest oli väga väike osa erasektorist vastanud.

Uuringu eesmärk oli ka välja selgitada, kas avalikule rakendusele koostatakse enam dokumentatsiooni kui kinnisele rakendusele. Vaatame esmalt arendajate küsimustiku tulemusi.

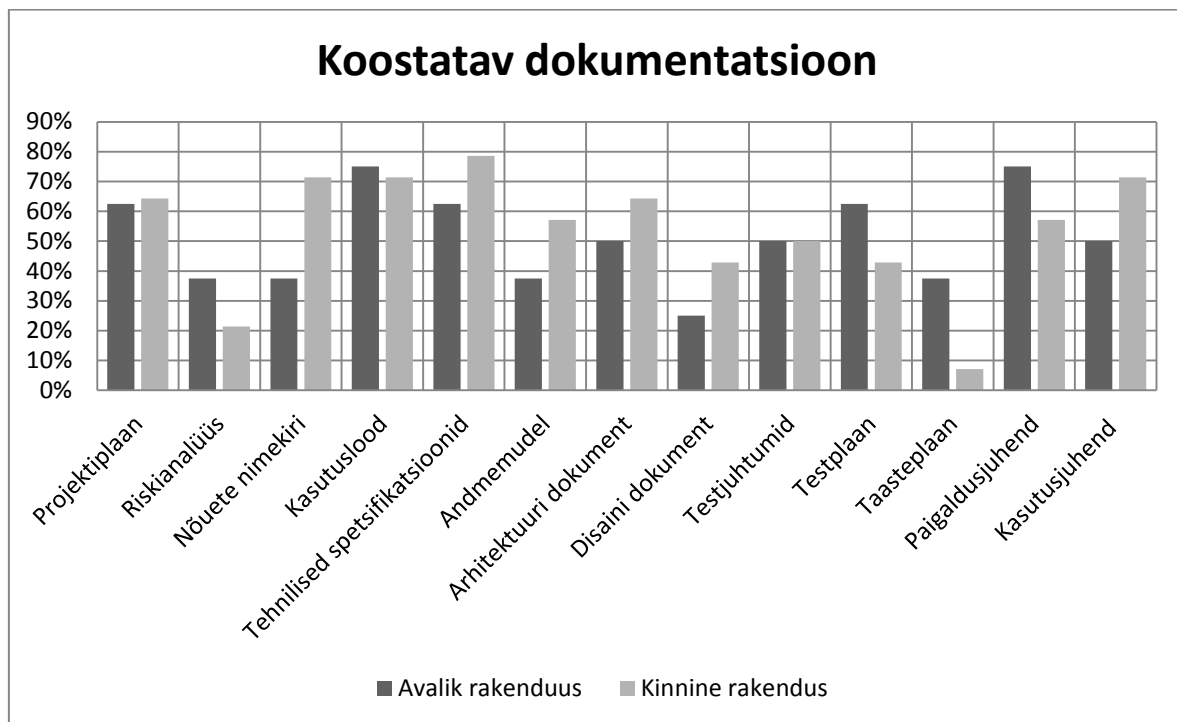


Diagramm 16, Koostatav dokumentatsioon avalikule ja kinnisele rakendusele. Arendaja küsimustiku vastused

Diagrammilt 16 on näha, et kõige suurem erinevus on kasutusjuhendi koostamises, 79% kinniste rakenduste arendajatest koostab kasutusjuhendi, kui ainult 38% avaliku rakenduse kasutajatest koostab selle. Vastupidiselt kasutusjuhendile on erinevus taasteplaani koostamises, mida koostatakse rohkem avalikele rakendustele. 38% avalike rakenduste ja 7% kinniste rakenduste arendajatest koostab taasteplaani.

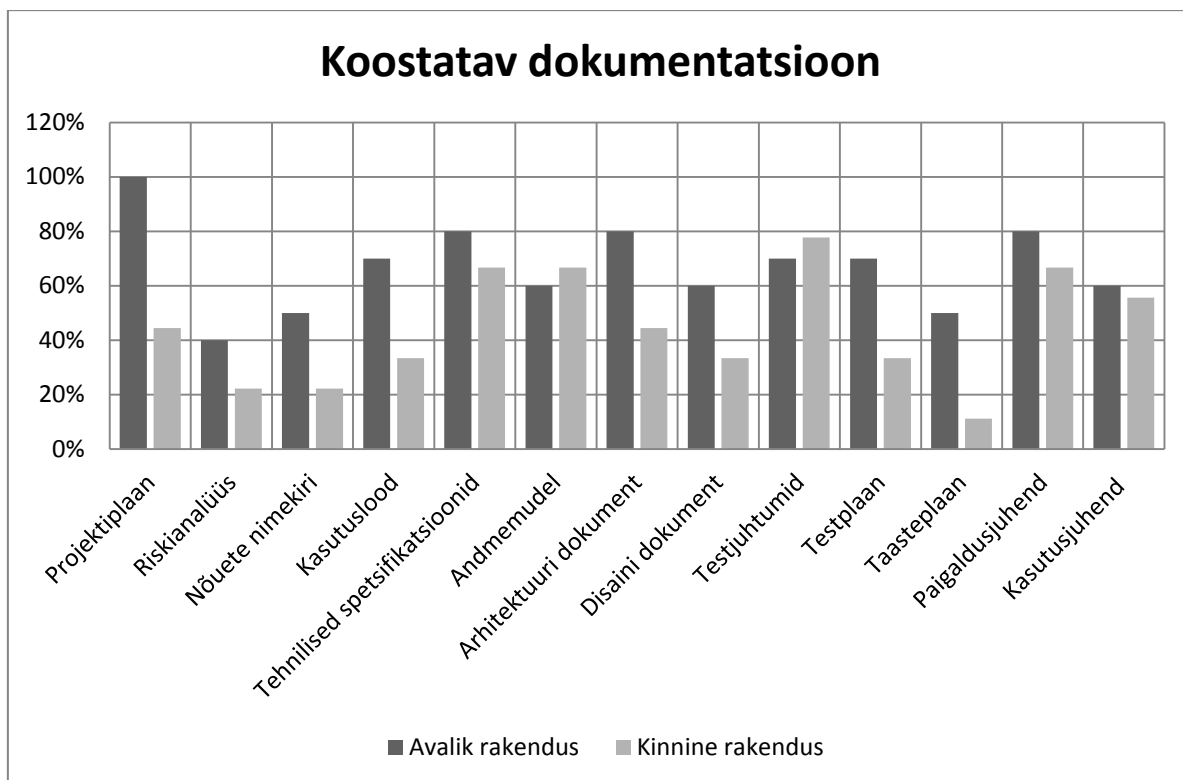


Diagramm 17, Koostatav dokumentatsioon avalikule ja kinnisele rakendusele. Kliendi küsimustiku vastused

Klientide küsimustiku tulemustest selgub aga, et kõige suurem erinevus on projektiplaani koostamises, mida kõik vastanud avaliku rakenduse tellijad ning ainult 44% kinnise rakenduse tellijad nimetasid. Diagramm 17 annab ülevaate kliendi küsitluse tulemustest avaliku ja kinnise rakenduse tellijate koostatavast dokumentatsioonist.

Klientide ja arendajate küsimustikus oli küsimus süsteemi kasutajate arvu kohta, et sarnaselt eelmistele küsimustele uurida, kas see mõjutab koostatavat dokumentatsiooni. Küsitluse tulemused sorteeriti kasutajate arvu järgi ning uuriti, kuidas mõjutab suurenev kasutajate arv dokumentide hulka. Suuri erinevusi dokumentatsioonis vastavalt kasutajate arvule ei leitud.

Järgmisena vaadeldakse arenduse kestvust ja selle mõju dokumentatsioonile. Vastanud ettevõtted jagatakse projektide kestvuse järgi kahte rühma; kuni 3 aastat ja vähemalt 3 aastat kestnud projektid.

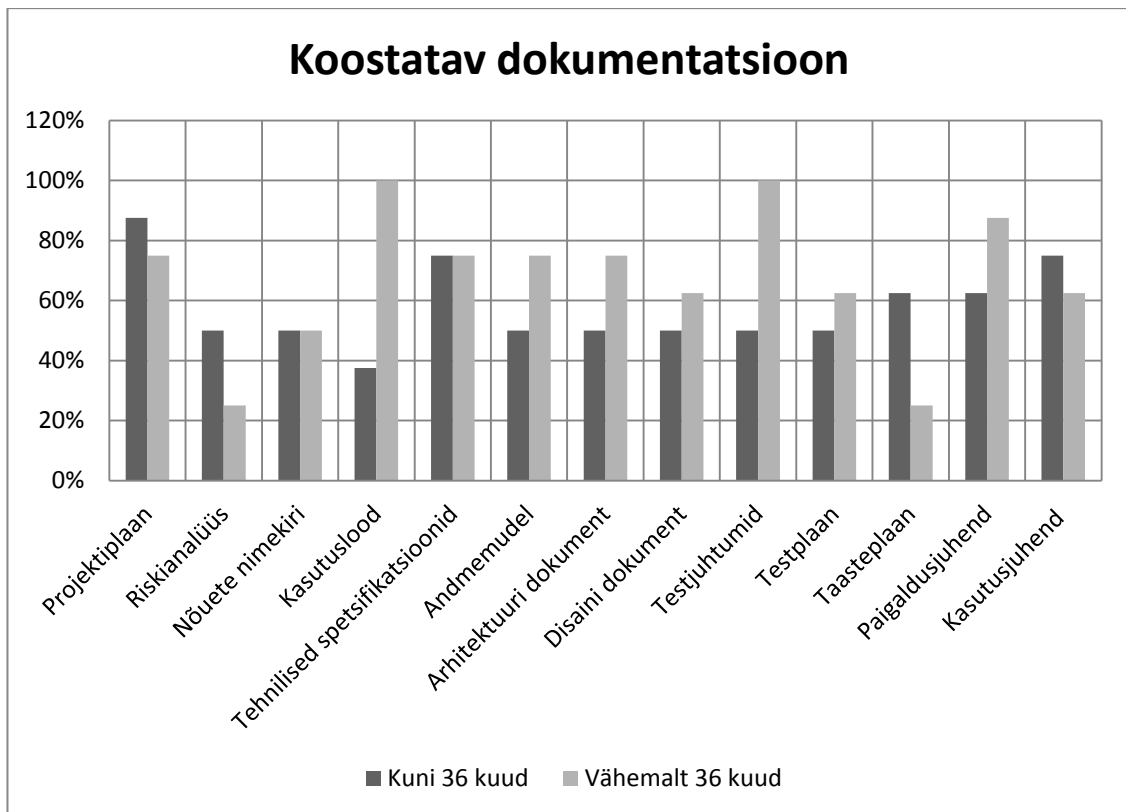


Diagramm 18, Koostatav dokumentatsioon projekti kestvuse järgi

Diagrammilt 18 on näha, et lühema ajaliselt arendatud projektides koostatakse rohkem dokumentatsiooni, mis on üllatuslik, kuna võiks arvata, et pikema ajaliselt arendatavas projektis vajatakse dokumente enam. Uurimusest selgub, et riskianalüüsi ja taasteplaani koostatakse palju rohkem alla 36 kuu kestnud projektide ja üle 36 kestnud projektides on nende dokumentide koostamine väga väikse arvuline.

Viimase projekte iseloomustava tegurina vaadeldakse äriloogika keerukust, mille kohta oli küsimus klientide küsimustikus. Projektid jagatakse uurimuses nelja rühma vastavalt äriloogikale: väga keeruline, keeruline, lihtne, väga lihtne äriloogika.

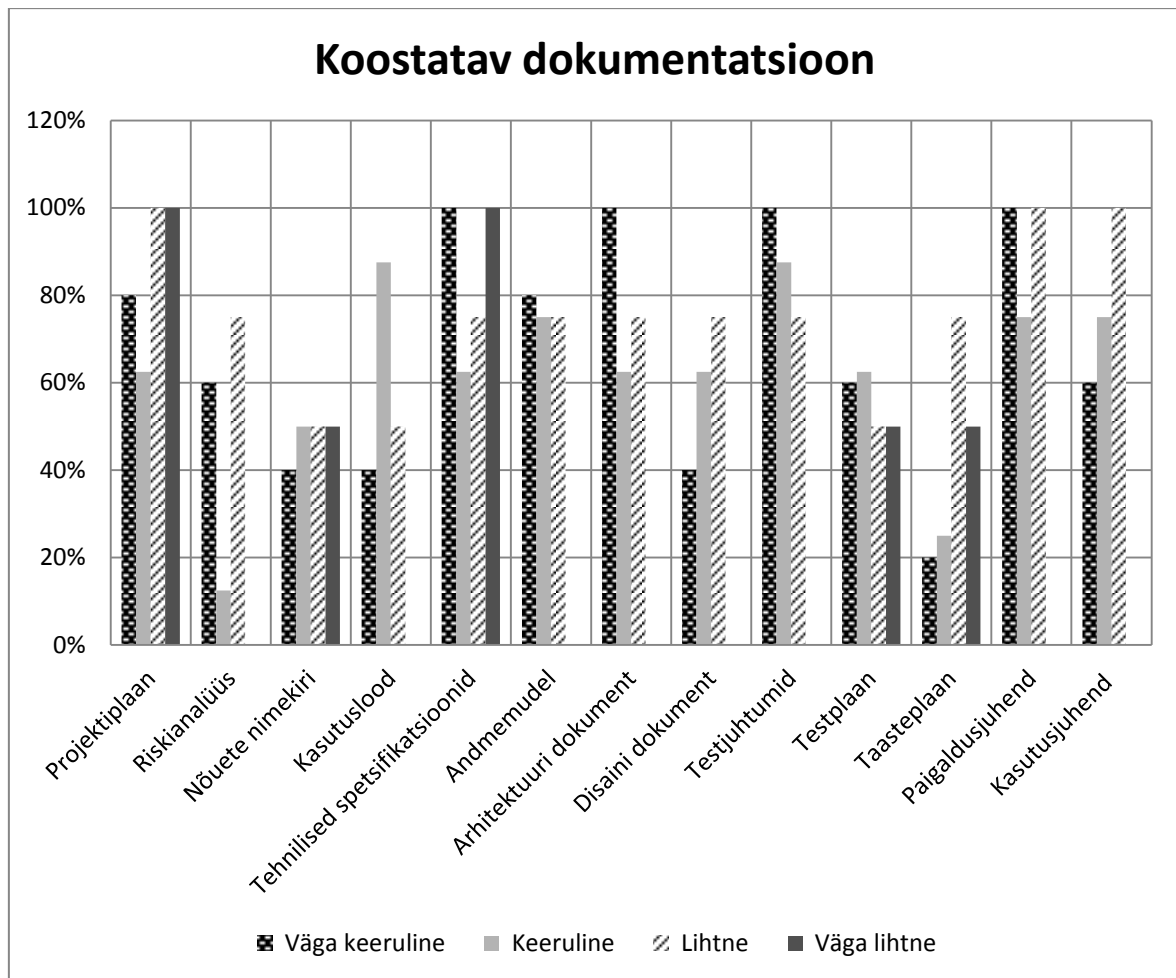


Diagramm 19, Koostatav dokumentatsioon ärioloogika keerukuse järgi jaotatult

Diagrammilt 19 on näha, et kõiki dokumendi liike ei mõjuta ärioloogika keerukus sarnaselt. Näiteks arhitektuuri dokumenti koostatakse enamuses projektides välja arvatud väga lihtsates, seejuures kõikides projektides, mille ärioloogika on väga keeruline. Testjuhtumitega on sarnaselt arhitektuuri dokumendile, et väga lihtsa ärioloogika korral neid uuringu tulemuste põhjal ei koostata. Lisaks on testjuhtumite tulemustest näha, et nende koostamise osakaal väheneb koos ärioloogika keerukusega, see tähendab kõigis väga keerulise ärioloogikaga projektidest, 88% keerulise ärioloogika projektides ning 75% lihtsa ärioloogikaga projektides koostatakse neid.

Kokkuvõtteks tuleb lisada, et kogu analüüs põhineb ainult küsitlusele vastanud tarkvara arendusega tegelevate ja tarkvara tellivate ettevõtete vastustel.

6 Uurimuse võrdlus

Parema ülevaate andmiseks uurimuse tulemustest võrreldakse seda internetist leitud uurimusega.

Uurimus viidi läbi 2002. aasta aprillis Ottava ülikoolis kasutades veebipõhist küsitlust [25]. Tegemist oli pika küsimustikuga, mis sisaldas 50 erinevat küsimust, mõned nendest on sarnased käeoleva töö raames Eestis läbi viidud küsimustega. Esimene küsimus oli koostatava dokumentatsiooni tüüpide kohta. Küsimuses anti ette järgmised valikuvariandid:

- Nõuete dokument
- Spetsifikatsioonid
- Disaini dokument
- Arhitektuuri dokument
- Testimise/kvaliteediga seotud dokumendid

Kõige suurem osa vastanutest, 73% vastanutest märkis, et koostavad spetsifikatsioone ja Eestis tehtud uurimuse vastused on väga sarnased, 73% arendajatest ja 79% klientidest koostab samuti spetsifikatsioone. Kõige väiksem osa 52% vastanutest märkis, et koostavad disaini dokumente, autori poolt tehtud küsimustikust selgus sarnaselt, et disaini dokumendi koostamise osakaal on väike, 36% arendajatest ja 53% klientidest vastasid, et koostavad seda. Kõige vähem koostatakse Eestis taasteplaani, mis dokumendiliigi valikut 2002. aastal läbi viidud küsitluses ei olnudki.

Järgmine küsimus, mis oli Ottava ülikoolis tehtud uurimuses sarnane Eestis läbi viidud uurimusega puudutas dokumentatsiooni kaasajastamist: „Kui kaua aega võtab dokumentatsiooni uuendamine, kui muudetakse tarkvara?“. Küsimuse tulemustest selgus, et enamasti tarkvaraarendajad uuendavad nõuetega seotud dokumente, spetsifikatsioone, disaini ja arhitektuuri dokumente harva, kuid testimise ja kvaliteediga seotud dokumente tihedamini, umbes paari päeva jooksul pärast muudatusi. Kui võrrelda seda töö käigus tehtud uurimuse küsimuse tulemustega, milliseid dokumendiliike kaasajastatakse, siis sellest ei tulnud välja, et testimisega seotud dokumente rohkem kaasajastatakse. Samas kuna dokumente kaasajastatakse väiksemal määral kui neid koostatakse, siis sellest saab järeldada, et seda tehakse tõesti harva, nagu näitas ka uuring.

Kokkuvõtteks andis Ottawa ülikoolis läbi viidud küsimustiku tulemustega võrdlus kinnitust töö raames Eestis läbi viidud uurimuse tulemustele. Koostatava dokumentatsiooni osa on vastavalt uurimuse tulemustele sarnane ning dokumentatsiooni kaasajastamise kohta puudus detailne info 2002. aastal läbi viidud küsitluses, kuid üldiseid järeldusi võimaldas see siiski teha. Kahjuks puudus info varasemas uurimuses dokumentatsiooni koostamise ja süsteemi eripärade vahel.

Kokkuvõte

Tarkvaraarenduse käigus loodav dokumentatsioon on olulisel kohal, kuna on nii kommunikatsioonivahend, informatsiooniallikas, kui juhend süsteemi kasutamiseks. Teiselt poolt pole üheselt selge, milline dokumentatsioon täidab kõiki eelnevaid rolle piisavalt, kui palju on vaja dokumentatsiooni arendaja või kliendi jaoks, et see oleks parim lahendus. Samuti kui suures mahus peab kaasajastama dokumentatsiooni või kas üldse peabki kõiki dokumente uuendama.

Töös vaadeldi dokumentatsiooni, mida koostatakse tarkvaraarendamise käigus. Kirjeldati, millised on erinevad dokumendid, mida võidakse arenduse käigus kirjutada. Järgmisena vaadeldi iga dokumendi kohta, kes on selle sihtgrupp ehk kellele on dokument kirjutatud, kas arendaja või kliendi meeskonna liikmetele või hoopis kasutajale. Kõiki dokumente ei kirjutata tarkvaraarenduse protsessi alguses, vaid erinevates etappides. Järgmises alapeatükis kirjeldati ülevaatlikult arenduse protsessi erinevaid etappe ning milliseid dokumente nende käigus koostatakse. Esimese peatüki viimases osas käsitleti dokumentatsiooni haldamise teemat, anti ülevaade erinevatest dokumentatsiooni haldamise tarkvaradest. Töö teises osas anti ülevaade, milline dokumentatsioon on kohustuslik ning mis seda mõjutab. Vaadeldi süsteemi eripärasid, mis võivad muuta ühe dokumendi selle süsteemi jaoks kohustuslikuks ja teise jaoks ebavajalikuks. Lisaks kirjeldati paari standardit, et anda ülevaade, nendes kirjeldatud soovitusliku dokumentatsiooni kohta. Kolmas peatükk andis ülevaate dokumentatsiooni kaasajastamise teemast. Kirjutati dokumendi uuendamisest, millal seda on tarvis teha, kuidas jne. Järgmises alapeatükis kirjeldati dokumentide versioneerimist. Töö viimane osa oli uurimuse kohta. Esmalt kirjeldati, kuidas küsitlust ettevalmistati ja kuidas see Eesti tarkvara arendajate ja tellijate seas läbi viidi. Järgmisena anti ülevaade arendaja vastustest küsimustikule ning seejärel kliendi vastustest. Peale tulemuste kirjeldamist analüüsiti vastuseid ja võrreldi varasemalt läbi viidud uurimuse tulemustega.

Töö käigus jõuti järeldusele, et Eesti tarkvaraarendajad koostavad kõiki küsitluses välja pakutud dokumente:

1. Projektiplaan
2. Riskianalüüs
3. Nõuete nimekiri

4. Kasutuslood
5. Tehnilised spetsifikatsioonid
6. Andmemudel
7. Arhitektuuri dokument
8. Disaini dokument
9. Testjuhtumid
10. Testplaan
11. Taasteplaan
12. Paigaldusjuhend
13. Kasutusjuhend

kuigi neid kõiki kliendid ei vaja. Samas on dokumente, mida arendajad koostavad vähem ja kliendid näevad selle järgi suuremat vajadust. Kaasajastamise küsimustest selgus, et leidub dokumente, mis on tellija seisukohast olulised, kuid väike osa arendajatest uuendab neid. Süsteemi eripärad mõjutavad koostatavat dokumentatsiooni väga erinevalt ning selgeid sõltuvusi kindla dokumendiliigi osas ei tulnud uurimusest välja.

Antud valdkonnas võiks järgmisena teha uurimuse Eestis iga dokumendi liigi kohta eraldi ning uurida sügavamaid põhjusi, miks koostatakse või ei koostata, kaasajastatakse või ei kaasajastata dokumente.

DOCUMENTATION IN IT SYSTEM

Master's Thesis

Elen Kivi

Summary

All large software companies write documentation. All software documents are not writing in the beginning of developing, documentation writing is continuous process, so in every software engineering phase could be writing documents. It is also different the audience. Some document's audience is developer's team, other client's team or users. As the amount of documentation is on the rise, it is important to identify what types of documents to include and what not. Likewise updating documentation is critical, especially when the number of updates is large.

The first part of the present research gave an overview about the types of documentation software engineers should write. The next part described each document's type audience groups and developing phase when each documents to be written. In the last part of first chapter wrote about documentation managing. The second chapter of the research is devoted to discussing which documents developer must write and what affect it. The third part of the research gave an overview how frequent software documentation should be updated. After that described how should versioning documents. The last part described how the inquiry was prepared and carried out. Inquiry was carried out in Estonia, Estonian software engineers and their clients participated. After that described result of survey and provided an analysis of its results.

The main goal of the present research was to give a detailed overview about software documentation, principles/strategies of how to write it, and the standards of such documentation. In addition, it provides an analysis of the matter concerning Estonian developers and clients. Finally, the research aims to figure out what types of documents engineers writes and should update and what IT system's property affect it.

The conclusion of the research that all software developers write documents which were in questionnaire:

1. Project plan
2. Risk analysis
3. Requirements document
4. Use case
5. Technical specifications
6. Data model
7. Architecture document
8. Design document
9. Test case
10. Test plan
11. Restore plan
12. Installation instruction
13. User manual

Besides, it is worth noting that clients seem not to perceive the importance of all such documents, as a small fraction answered that some of them are not important. Finally, there are some types of documents that clients consider very important that are, however, kept updated by a few engineers.

As the result of the present research we may say that systems' idiomaticness like:

1. Public or private IT system
2. The amount of users
3. The complexity of business logic
4. Duration of project

affect which documents developers write but direct dependency concrete document and idiomaticness didn't find from survey.

Within this area of research, further studies should be done to find out the reasons why developers write or do not write, update or do not update these types of documents in Estonia.

Viited

1. <http://www.itl.nist.gov/fipspubs/fips105.pdf> "Guideline for software documentation management"

Viimati vaadatud 05.04.2011

2. www.site.uottawa.ca/~tcl/gradtheses/aforward/

Software Documentation – Building and Maintaining Artefacts of Communication

Viimati vaadatud 11.05.2011

3. http://books.google.ee/books?id=ZD4EAAAAMBAJ&pg=PT8&lpg=PT8&dq=software+documents++1980&source=bl&ots=OrHBoUTl3F&sig=O2LS86lXX51XqA_HXofyeMAA9Jw&hl=et&ei=rzfCTfv-LJDRsgaOw8GSAQ&sa=X&oi=book_result&ct=result&resnum=9&ved=0CFcQ6AEwCA#v=onepage&q&f=false

Info world, "You et What You Pay for" Hank Levine 1980

Viimati vaadatud: 05.05.2011

4. www.siseministeerium.ee/failid/Projplaani_koostamine.rtf, "Projektiplaani koostamine", Algis Perens

Viimati vaadatud 18.05.2011

5. <http://www.projectsmart.co.uk/pdf/project-planning-step-by-step.pdf> "Project Planning a Step by Step Guide", Duncan Haughey

Viimati vaadatud 18.05.2011

6. <http://dspace.utlib.ee/dspace/bitstream/handle/10062/579/loogkaspar.pdf;jsessionid=CDB8B920F43F178A4EE8DEAEE738AB47?sequence=5> "

Tarkvara arenduseprotsess erp-tarkvara juurutusprojektides", Kaspar Loog, 2005

Viimati vaadatud 18.05.2011

7. <http://blog.slickedit.com/2007/05/how-to-write-an-effective-design-document/> "How to Write an Effective Design Document", Scott Hackett, mai 2007

Viimati vaadatud 18.05.2011

8. <http://arxiv.org/ftp/arxiv/papers/1005/1005.0595.pdf> “Software Design Document, Testing, and Deployment and Configuration Management”, Mr. Serguei Mokhov, talv 2010

Viimati vaadatud 18.05.2011

9. <http://www.ics.uci.edu/~wscacchi/Papers/SE-Encyc/Process-Models-SE-Encyc.pdf> “Process Models in Software Engineering”, Walt Scacchi, Institute for Software Research, University of California, Irvine, veebruar 2001

Viimati vaadatud 18.05.2011

10. <http://www.shellmethod.com/refs/SDLC.pdf> “The Software Development Life Cycle (SDLC) For Small To Medium Database Applications”

Viimati vaadatud 18.05.2011

11. <http://www.literateprogramming.com/documentation.pdf> “Software documentation”, Ian Sommerville, Lancaster University, UK 21.11.2010

Viimati vaadatud 18.05.2011

12. www.tlu.ee/~coolbeck/Files/Tarkvaratehnika/tarkvara.ppt “Tarkvara testimine” Jevgeni Boldõšev, Anastassia Bõtšenkova

Viimati vaadatud 18.05.2011

13. http://books.google.ee/books?id=K7n52MOIDQsC&pg=PA466&dq=software+documents++1981&hl=en&ei=iUDCTeGfO4npOYGHxJ0I&sa=X&oi=book_result&ct=result&resnum=5&ved=0CG8Q6AEwBA#v=onepage&q=software%20documents%20%201981&f=false Software reliability handbook, Paul Rook 1990

Viimati vaadatud: 05.05.2011

14. <http://www.aiim.org/What-is-Document-Management> “What is document management”

Viimati vaadatud: 16.04.2011

15. <http://www.hariduskeskus.ee/opiobjektid/grupitoo/?%D5PPEMATERJALID:Dokumendihaldustarkvara>, Pärnumaa Kutsehariduskeskus, Varje Tipp
Viimati vaadatud: 05.05.2011
16. <http://postipoiss.microlink.ee/>, "Postipoisi tutvustus"
Viimati vaadatud 05.05.2011
17. http://microfon.microlink.ee/index.php?option=com_content&task=view&id=46&Itemid=45 "Opentext LiveLink Sampo Pangas dokumente ja teavet haldamas"
Viimati vaadatud 05.05.2011
18. http://www.cs.tlu.ee/instituut/opilaste_tood/bakalaureuse_ja_diplomitood/2004_kevad/Triinu_Gross/Triinu_Gross_Bakalaureuse_Too.pdf "Tarkvara arenduseprotsessi üldpõhimõtted TPÜ-s", Triinu Gross, 2004
Viimati vaadatud 18.05.2011
19. http://www.lap.ttu.ee/erki/failid/konspekt/tarkvara_kvaliteet_ja_standardid_idx5721/idx5721_loengukonspekt.pdf, "Tarkvara kvaliteet ja standardid" 1994
Viimati vaadatud 18.05.2011
20. <http://www.cs.tlu.ee/~inga/progbaas/Materjalid/tks-loeng.pdf> "Tarkvara kvaliteet ja standardid", Jaak Tepandi, 2007
Viimati vaadatud 18.05.2011
21. <http://www.techrepublic.com/article/get-it-done-keep-documentation-updated-with-change-management/1052001> "Get IT Done: Keep documentation updated with change management", Allen V., 2002
Viimati vaadatud 18.05.2011
22. <http://www.agilemodeling.com/essays/agileDocumentation.htm#UpdatingDocumentation> "Agile/Lean Documentation: Strategies for Agile Software Development" Scott W. Amber 2006
Viimati vaadatud 18.05.2011

23. http://www.infracore.com/products/silva/docs/author_advanced/version_management_docs "Document versioning"

Viimati vaadatud 18.05.2011

24. <http://www.northumbria.ac.uk/static/5007/uso/vercon.pdf> "Document Versioning"

Northumbria University Records Management

Viimati vaadatud 19.03.2011

25. www.site.uottawa.ca/~aforward/thesis/aforward_appendix_A-E.doc

Software Documentation, Andrew Forward, 2002

Viimati vaadatud: 05.05.2011

Lisad

Lisa 1, küsimustik arendajale

Dokumentatsiooni koostamine tarkvara ettevõttes

Palun täita järgnev ankeet

Kas Teie klient on riiklik asutus?

☐ Jah

☐ Ei

Kas tegemist on avaliku rakendustega?

☐ Jah

☐ Ei

Kui palju kasutajaid on rakendusel?

Milliseid dokumente koostate?

- ☐ Projektiplaan
- ☐ Riskianalüüs
- ☐ Nõuete nimekiri
- ☐ Kasutuslood
- ☐ Tehnilised spetsifikatsioonid
- ☐ Andmemudel
- ☐ Arhitektuuri dokument
- ☐ Disaini dokument
- ☐ Testjuhtumid
- ☐ Testplaan
- ☐ Taasteplaan
- ☐ Paigaldusjuhend
- ☐ Kasutusjuhend
- ☐ Muu

Milliseid dokumente peate vajalikeks neist?

- ☐ Projektiplaan
- ☐ Riskianalüüs
- ☐ Nõuete nimekiri
- ☐ Kasutuslood
- ☐ Tehnilised spetsifikatsioonid
- ☐ Andmemudel
- ☐ Arhitektuuri dokument

**Milliseid dokumente kaasajastate pidevalt?
(up-to-date)**

- ☐ Disaini dokument
- ☐ Testjuhtumid
- ☐ Testplaan
- ☐ Taasteplaan
- ☐ Paigaldusjuhend
- ☐ Kasutusjuhend
- ☐ Muu

- ☐ Projektiplaan
- ☐ Riskianalüüs
- ☐ Nõuete nimekiri
- ☐ Kasutuslood
- ☐ Tehniline spetsifikatsioon
- ☐ Andmemudel
- ☐ Arhitektuuri dokument
- ☐ Disaini dokument
- ☐ Testjuhtumid
- ☐ Testplaan
- ☐ Taasteplaan
- ☐ Paigaldusjuhend
- ☐ Kasutusjuhend
- ☐ Muu

Miks kaasajastate neid dokumente?

Lisa 2, küsimustik kliendile

Dokumentatsioon tarkvaraarenduses

Palun täitke järgnev ankeet!

Kas tegemist on avaliku rakendusega?

- ☐ Jah
- ☐ Ei

Kui palju on rakendusel kasutajaid?

Järgnevad küsimused on arendatava süsteemi kohta.

Kui kaua on kestnud rakenduse arendamine?

Kuidas hindate rakenduse ärioloogikat?

- ☐ Väga keeruline
- ☐ Keeruline
- ☐ Lihtne
- ☐ Väga lihtne

Milliseid dokumente koostab arendaja?

- ☐ Projektiplaan
- ☐ Riskianalüüs
- ☐ Nõuete nimekiri
- ☐ Kasutuslood
- ☐ Tehnilised spetsifikatsioonid
- ☐ Andmemudel
- ☐ Arhitektuuri dokument
- ☐ Disaini dokument
- ☐ Testjuhtumid
- ☐ Testplaan
- ☐ Kasutusjuhend
- ☐ Taasteplaan
- ☐ Paigaldusjuhend
- ☐ Muu

Millised dokumendid on neist ebavajalikud?

- ☐ Projektiplaan
- ☐ Riskianalüüs
- ☐ Nõuete nimekiri
- ☐ Kasutuslood
- ☐ Tehnilised spetsifikatsioonid
- ☐ Andmemudel
- ☐ Arhitektuuri dokument
- ☐ Disaini dokument
- ☐ Testjuhtumid
- ☐ Testplaan
- ☐ Kasutusjuhend

Milliseid dokumente peaks veel arendaja koostama?

- ☐ Taasteplaan
- ☐ Paigaldusjuhend
- ☐ Muu
-

- ☐ Projektiplaan
- ☐ Riskianalüüs
- ☐ Nõuete nimekiri
- ☐ Kasutuslood
- ☐ Tehnilised spetsifikatsioonid
- ☐ Andmemudel
- ☐ Arhitektuuri dokument
- ☐ Disaini dokument
- ☐ Testjuhtumid
- ☐ Testplaan
- ☐ Kasutusjuhend
- ☐ Taasteplaan
- ☐ Paigaldusjuhend
- ☐ Muu

Kuidas hindate dokumentide kaasajastamist ehk dokumentatsiooni uuendamist?

- ☐ Piisav, vajalik dokumentatsioon on kaasajastatud
- ☐ Keskpärane, on puudujääke
- ☐ Kasiin, palju vajalikku dokumentatsiooni on aegunud

Lisa 3, küsimustiku tulemused

Arendajate vastused

Küsimus		Vastused
Kas Teie klient on riiklik asutus?	Jah	11
	Ei	11
Kas tegemist on avaliku rakendustega?	Jah	8
	Ei	14
Kui palju kasutajaid on rakendusel?	---	Keskmine 11 400
Milliseid dokumente koostate?	Projektiplaan	13
	Riskianalüüs	6
	Nõuete nimekiri	12
	Kasutuslood	16
	Tehnilised spetsifikatsioonid	15
	Andmemudel	11
	Arhitektuuri dokument	12
	Disaini dokument	7
	Testjuhtumid	10
	Testplaan	10
	Taasteplaan	4
	Paigaldusjuhend	13
	Kasutusjuhend	13
	Muu	
Milliseid dokumente peate vajalikeks neist?	Projektiplaan	12
	Riskianalüüs	7
	Nõuete nimekiri	12
	Kasutuslood	15
	Tehnilised spetsifikatsioonid	15
	Andmemudel	9
	Arhitektuuri dokument	15
	Disaini dokument	8
	Testjuhtumid	12
	Testplaan	11
	Taasteplaan	5
	Paigaldusjuhend	14
	Kasutusjuhend	13
	Muu	
Milliseid dokumente kaasajastate pidevalt? (up-to-date)	Projektiplaan	7
	Riskianalüüs	3
	Nõuete nimekiri	5
	Kasutuslood	7
	Tehnilised spetsifikatsioonid	7
	Andmemudel	5
	Arhitektuuri dokument	5
	Disaini dokument	2
	Testjuhtumid	6

	Testplaan	5
	Taasteplaan	4
	Paigaldusjuhend	7
	Kasutusjuhend	10
	Muu	
Miks kaasajastate neid dokumente?	---	

Klientide vastused

Küsimus		Vastused
Kas Teie organisatsioon on riiklik asutus?	Jah	15
	Ei	4
Kas tegemist on avaliku rakendustega?	Jah	10
	Ei	9
Kui palju kasutajaid on rakendusel?	---	Keskmine 2 300
Kui kaua on kestnud rakenduse arendamine?	---	Keskmine 42 kuud
Kuidas hindate rakenduse äriloogikat?	Väga keeruline	5
	Keeruline	8
	Lihtne	4
	Väga lihtne	2
Milliseid dokumente koostate?	Projektiplaan	15
	Riskianalüüs	7
	Nõuete nimekiri	9
	Kasutuslood	11
	Tehnilised spetsifikatsioonid	15
	Andmemudel	13
	Arhitektuuri dokument	13
	Disaini dokument	10
	Testjuhtumid	15
	Testplaan	11
	Taasteplaan	7
	Paigaldusjuhend	15
	Kasutusjuhend	13
	Muu	
Millised dokumendid on neist ebavajalikud?	Projektiplaan	
	Riskianalüüs	
	Nõuete nimekiri	
	Kasutuslood	1
	Tehnilised spetsifikatsioonid	
	Andmemudel	1
	Arhitektuuri dokument	
	Disaini dokument	1

	Testjuhtumid	
	Testplaan	
	Taasteplaan	1
	Paigaldusjuhend	1
	Kasutusjuhend	
	Muu	
Milliseid dokumente peaks veel arendaja koostama?	Projektiplaan	
	Riskianalüüs	2
	Nõuete nimekiri	
	Kasutuslood	
	Tehnilised spetsifikatsioonid	
	Andmemudel	1
	Arhitektuuri dokument	2
	Disaini dokument	3
	Testjuhtumid	2
	Testplaan	2
	Taasteplaan	1
	Paigaldusjuhend	
	Kasutusjuhend	1
	Muu	
Kuidas hindate dokumentide kaasajastamist ehk dokumentatsiooni uuendamist?	Piisav, vajalik dokumentatsioon on kaasajastatud	3
	Keskpärase, on puudujääke	14
	Kasin, palju vajalikku dokumentatsiooni on aegunud	2